Chapter 3
Temporal Relations

> The habit of looking to the future and thinking that the whole meaning of the present lies in what it will bring forth is a pernicious one. There can be no value in the whole unless there is value in the parts.
>
> *Conquest of Happiness*
> BERTRAND RUSSELL

3.1 Introduction

Having discussed timex and events in the previous chapter, we move on to discuss the temporal relations that exist between them. This chapter briefly describes temporal relations and surveys the state of the art in automatic temporal relation annotation. Extra attention is given to prior work on temporal relation typing. We will discover that temporal link typing remains a difficult problem, despite multiple sophisticated approaches. The overall picture highlights persistent difficulties in temporal relation typing and suggests that to understand how to temporally order events described in text, we need to draw upon multiple heterogeneous information sources.

Time can be described as a constantly progressing sequence of events. This sequential attribute is critical to the concept of a timeline, on which one may place events. Absolute locations upon the timeline are described using timexes. Conversely, event positions are not be absolute and sometimes can be temporally situated only in terms of their relation to other events or to timexes. This means that correctly identifying the temporal relations between pairs made up of events or timexes is critical to automatic processing of time in language.

In terms of information extraction, we are interested in either assigning an absolute temporal value to the start and end points of temporal entities, or describing these points in terms of other entities. It is helpful to have at least one value firmly anchored – normalised – to a timeline. If we have a specific distance between two events and the position of one has already been normalised, it is trivial to also normalise the other; for example, in "*John was born on the 24th April, 1942. His mother left the hospital nine days later.*", we have a "*born*" event which is already anchored and a "*left*" event which we can attach to 3nd May, 1942 with some inference.

L.R.A. Derczynski, *Automatically Ordering Events and Times in Text*,
Studies in Computational Intelligence 677, DOI 10.1007/978-3-319-47241-6_3

In cases where normalisation is not immediately possible, however, we may mark a relation between two events using a temporal link. This allows the representation of non-absolute temporal information. A network of events, times and relations help one to determine the temporal arrangement of events described in discourse.

While events and times are overt, the temporal relations that exist between them are abstract. Events and times in a text have lexicalised representations, but the ordering of them is not always made explicit. This contributes to the difficulty of temporal relation identification and typing.

The problem of reasoning about and of representing temporal information has been addressed in the fields of knowledge representation and artificial intelligence. Once a representation has been defined, we may formally describe certain temporal structures within a discourse and start to make inferences about temporal relations. Temporal relation types expressed in language do not necessarily match the classes available in an annotation schema. However, to perform automatic temporal relation extraction, it is important to decide a set of temporal relations. Part of the purpose of fixing this relation set is to aid inference; another is to provide a stable framework for human annotation.

In this chapter, we will first define the concept of temporal relations. This is followed by an exploration of different sets of temporal relation types applicable to linguistic annotation. After this, we discuss ways of annotating temporal relations over discourse, and the concepts of relation folding, temporal closure and temporal annotation as a graph are introduced. Next, the chapter introduces the general problem of automatic temporal relation annotation. This is followed by a literature review, coming up to the state of the art in automatic temporal relation typing. Finally, the chapter concludes with an analysis of the state of the art and the automatic relation typing problem.

3.2 Temporal Relation Types

Temporal algebras and logics allow one to deduce relationships between events based on their connection to other times and events, using a set of rules. These rules depend on the specific set of event relationship types and a set of relation types. Interval, point and semi-interval logics are all available. Building on STAG (Sheffield Temporal Annotation Guidelines, [1, 2]), TimeML (Sect. 2.3.2.1) defines its own set of interval relations, based on Allen's interval algebra [3]; point-based algebra can be useful for rapid reasoning; semi-interval reasoning relaxes the burden of specification required when both points of an interval need to be found, in order to avoid over-specification when working with events described by natural language and are discussed in Sect. 3.2.3.

For the context of this book, interval algebrae are considered to be those that define types of relation between intervals and a set of axioms for operating with these relations; an **interval** has a start and an end point. Some temporal logics use points instead of intervals. For interval logics, a point event may be represented by an

interval whose start and end occur simultaneously; a **proper interval** is an interval where the end occurs after the start [4].

Temporal logics deal with reasoning about the relations that hold between intervals. Early examples of temporal logics include Prior's calculus for a modal tense logic calculus [5] and Bruce's model [6], which also includes axioms for event reasoning withing a temporal system.

This section first presents a few temporal interval algebrae, each with a specific purpose; finally, we will introduce the concept of temporal closure.

Applications of temporal logics can be found in multiple areas of computer science, including the verifying and testing time-sensitive parts of computer programs, in providing a temporal data representation for artificial intelligence systems and for representing temporal semantics in natural language processing. This section does not comprehensively discuss the full range of temporal logics, rather just those that deal with intervals and that have been previously applied to (or designed for) natural language processing. Other work has examined temporal logics in detail [7–9].

This section discusses some temporal interval algebras and their use in representing and reasoning over time as part of temporal information extraction. Firstly, there is a very minimal algebra, including just three relationship types. The limited number of potential relationship types makes it easier to visualise the relations between events and simpler to implement and troubleshoot problems that arise while reasoning. Secondly, we cover Allen's interval logic, which defines enough relations to cover all possible relations between a pair of temporal intervals. Finally is Freksa's logic based on semi-intervals, which tries to better capture and reason with the event relations pres in natural language discourse.

3.2.1 A Simple Temporal Logic

One can describe many basic relations between intervals using just three relations - BEFORE, INCLUDES and SIMULTANEOUS. If we encounter something such as *I washed after cleaning the sewer*, if events are denoted as E we can have simply reverse argument order to have $E_{cleaning}$ BEFORE E_{wash}. As part of a larger investigation into temporal reasoning on information found in discourse, [10] introduces a minimal logic based on three simple relations than only requires ten rules for temporal inference. The simplicity of this system makes it both easy to implement and easy to think about. However, the set of just three relations is small and the temporal relations expressed in natural languages can be more precisely represented using a wider set of temporal relation types. For example, if two intervals overlap but do not share any start or end points (such as winter in the northern hemisphere, which may begin in a November, and a calendar year), neither before, includes or simultaneous is precise enough to describe their temporal relation.

3.2.2 *Temporal Interval Logic*

Allen's interval logic [3] describes a set of temporal relations that may exist between any event pair. Allen introduces the concept of events (represented as intervals) as nodes in a graph, where the edges connecting nodes represent a relationship between two intervals. Where it is not clear that a single type of relation should exist between a pair of events, a disjunction of all possible relationship types is used to label the connection edge. Further, Allen provides an algorithm for deducing relationships between previously unconnected nodes.

The relations are listed in Table 3.1. Each of these gives a specific configuration of interval start and end points. Based on this, a transitivity table is provided for inferring new relations between intervals that hold common events. A full transitivity table is given in Table A.9.

A story typically describes more than one event, with some temporal ordering. Example 4 describes two events, setting out (E1) and living happily (E2).

Example 4 Little Red Riding Hood set out to town. She lived happily ever after.

The temporal link here is that she lived happily after setting out, signalled by both the textual order and also the use of the word *after*. Now, we can define a temporal link that says E2 AFTER E1 and label it L1.

It is improper to adventure without a cloak; perhaps we could introduce a new sentence in our text. See Example 5.

Table 3.1 Allen's temporal interval relations

Relation	Explanation of A-relation-B
BEFORE	Where A finishes before B starts
AFTER	Where A starts after B ends
DURING	Where A starts and ends while B is ongoing
CONTAINS	Inverse of DURING
OVERLAPS	Where A starts before B and ends during B
OVERLAPPED- BY	Inverse of OVERLAPS
MEETS	Where A ends at the point B begins
MET- BY	Inverse of MEETS
STARTS	Where A and B share their start point, but A ends before B does
STARTED- BY	As starts, but B ends first
FINISHES	Where A and B share their end point, but A begins later (and is thus shorter)
FINISHED- BY	As finishes, but B is the shorter/younger interval
EQUAL	Where A and B start and end at the same time

Example 5 Little Red Riding Hood set out to town. She put on her cape before leaving. She lived happily ever after.

This suggests a new dressing event, E3, signified by *putting on*. We also know the link between our new event and E1, setting out; E3 BEFORE E1. We'll call this L2. The story can now be represented by 3-node graph (events E1, E2 and E3), with two labelled edges (L1 and L2).

- E1: setting out
- E2: living happily
- E3: put on cape
- L1: E2 AFTER E1
- L2: E3 BEFORE E1

A visual representation of the temporal graph of these events and links is given in Fig. 3.1. This current graph leaves the relation between E3 and E2 unspecified. Narrative convention and human intuition tell us that we should use a linear model of time and suggest that anything that happens before the girl sets out must also happen before her living happily ever after. In this case, we can formally describe that knowledge with rules:

$\forall x, y : x$ AFTER $y \rightarrow y$ BEFORE x

$\forall x, y, z : x$ BEFORE y, y BEFORE $z \rightarrow x$ BEFORE z

Thus, Little Red Riding Hood puts on her cape before living happily ever after and we can now introduce L3 as E3 BEFORE E2, completing the graph. This also describes BEFORE as a transitive relation.

Allen's logic was considered exciting because it was implementable at the time, unlike other temporal logics (e.g. [11]), and was also expressive; it has since been adopted by logicians, the verification and testing community and those interested in time in language. For a further review of temporal interval logics, one should see [8] and [12].

3.2.3 Reasoning with Semi-intervals

Temporal interval logic is not perfect. Determining consistency in any but the smallest scenarios quickly becomes intractable and is NP-hard [13, 14]. Problems arise when dealing with instantaneous events (e.g. "improper" intervals – Sect. 2.3); inconsistencies appear when events are allowed to have a duration of zero and the system is explicitly not structured to deal with these [15]. Semi-intervals are intervals where only one bound needs to be described (e.g. the start point or end point). It is contended that such relaxed definitions, when compared to fully-described intervals, can better represent the relations expressed in natural language. In this section, we discuss the shortcomings of temporal interval algebra and introduce a system for reasoning with semi-intervals.

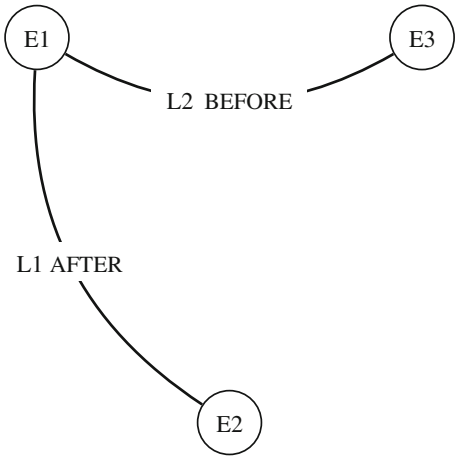

Fig. 3.1 Temporal graph of a simple story

Some common relation typing tasks are difficult to perform with interval relations. For example, newswire articles usually have a document creation time (DCT) or a publication date, which appears in document metadata and as a timex in the main body of discourse. They often contain at least a few events whose initiation is described in the past tense. In these cases, it is hard to determine whether an event's final bound stops at or continues past DCT, especially for states.

Example 6 contains an excerpt from a news report, uttered mid-way through a day. The timex *Today* has a specific meaning of a 24-h period. The start of the *control* event is unclear, but contextually we might assume that it begins before *Today*. Regardless of the arrangements of starting points of these two intervals, which could perhaps be discovered with further investigation, the arrangements of the endpoints of *Today* and *control* are unknowable at the time of utterance. Control could be relinquished before the day is over, at the precise end of the day, or later. This uncertainty makes it difficult to assign a relation from Allen's set to the two intervals. Without knowledge about the endpoints of these intervals, we can only say that the time-event relationship is one of *Today* $\langle overlapinverse, finishes, during \rangle$ *control*.

Example 6 Today, rebels still control the airfield and surrounding area.

To this end, [16] suggests a temporal algebra targeted at those dealing with natural language. It builds upon previous seminal work on logics that handle the uncertainties of time as described in language [17]. As long as we know that intervals begin before they end, we can start to describe relations between semi-intervals as disjunctions of Allen relations. It is quickly observed that particular Allen relations occur together, when dealing with incomplete knowledge about events. Freksa summarises these, defining terms for conceptual neighbours – "two relations between pairs of events are **conceptual neighbours** if they can be directly transformed into one another by continuously deforming (i.e. shortening, lengthening, moving) the events (in a topological sense)". For example, BEFORE and MEETS neighbour, as one can change the

relation between two events from one of these to the other by adjusting the endpoint of the interval that starts earliest. We then also have **conceptual neighbourhoods**, which are sequences of relations which are conceptual neighbours.

Freksa's system tackles uncertainty about knowledge linking two events and allows us to capture information from text that may not describe all intervals completely. Using groups of relations that commonly co-occur during inference, Freksa describes a temporal algebra, labelling certain groups of Allen relations as relations in their own right. The algebra specifies a transitivity table. The table is based on commonly co-occurring groups of relations.

For example, from Freksa's set, the relation A `older` B applies whenever A's start point happens before B's start point; no attention is paid to their endpoints and so any of A [BEFORE, IBEFORE, ENDED_BY, INCLUDES] B apply. From this example at least one instance in English where a semi-interval logic would be useful is immediately clear. Further examples are provided in Freksa's paper. Additionally, Sect. 6.4.2 investigates semi-interval logic in the context of tense-based temporal relation typing.

3.2.4 Point-Based Reasoning

As their name suggests, point-based temporal logics work only with the ordering of individual points and do not cater for the concept of an interval. They are less prone to the over-specification problem that full interval algebras have (see above). It is possible to decompose intervals to their beginning and end points. Only equality and precedence operators are needed to described binary relations between these points. Point-based algebrae can be very fast to process, a feature which tools such as SputLink [18] and CAVaT [19] exploit. They also better lend themselves to graph-based reasoning about temporal structures in text [20]. However, it is more complicated for humans to annotate using points instead of intervals and the semantics of temporal relations in text are better represented with interval or semi-interval labels. Because of these reasons and because temporal annotation is already a difficult and exhausting task for human annotators, point-based reasoning and temporal logics are generally restricted to the domain of fully automated reasoning [8].

3.2.5 Summary

We have outlined the requirements for temporal logic in the context of language and detailed examples; a simple 3-relation logic, Allen's interval logic, Freksa's semi-interval logic, and point-based reasoning. In the next section, we will see how using these logics with an existing document can tell us about temporal links that have not yet been annotated.

Table 3.2 TimeML temporal relations

Relation	Explanation of A-relation-B
BEFORE	A finishes before B starts
AFTER	A starts after B ends
INCLUDES	A start before and finishes after B
IS_INCLUDED	A happens between B's start and finish
DURING	A occurs within duration B
DURING_INV	A is a duration in which B occurs
SIMULTANEOUS	A and B happen at the same time
IAFTER	A happens immediately after B
IBEFORE	A happens immediately before B
IDENTITY	A and B are the same event/time
BEGINS	A starts at the same time as B, but finishes first
ENDS	A starts after B, but they finish at the same time
BEGUN_BY	A starts at the same time as B, but goes on for longer
ENDED_BY	A starts before B, but they finish at the same time

3.3 Temporal Relation Annotation

The work in this book primarily concerns temporal relation annotation using intervals, as opposed to points or semi-intervals. This section is about turning the abstract idea of temporal ordering into something well-defined that we can reason with directly – the process of annotation.

Temporal relations obtain between two endpoints. They describe the natural of a temporal relation between those endpoints. Those endpoints my be either times or events, and needn't be of the same type. Therefore, a temporal relation annotation must at the minimum specify two endpoints and a relation (or label describing the relation) that exists from the first to the second. Optionally, additional information may be included, such as pointers to phrases that help characterise the relation.

There are three sets of temporal relations commonly used for linguistic annotation: Allen's original set (Table 3.1), the TimeML interval relations (Table 3.2), and the TempEval-1 and TempEval-2 simplified set (Table 3.3).

The TimeML relations are intended to be interpreted slightly less strictly than the Allen set. As language is imprecise and there is often some uncertainty around the precise location of endpoints, a little variance is permitted; actual events need not

Table 3.3 The relation set used in TempEval and TempEval-2

Relation	Explanation of A-relation-B
BEFORE	Where A finishes before B starts
AFTER	Where A starts after B ends
OVERLAP	Where any parts of A and B co-occur
BEFORE- OR-OVERLAP	A disjunction of BEFORE and OVERLAP
OVERLAP- OR-AFTER	A disjunction of OVERLAP and AFTER
VAGUE	For completely underspecified relations

start and end at the exact same (e.g.) millisecond[1] – instead, interpretation is left to the annotator.

TimeML describes realis, non-aspectual temporal relations using the **TLINK** element. The TLINK element's **relType** attribute's value is that of the temporal relation's type.

3.3.1 Relation Folding

Many of the relations used in both TimeML and Allen's interval algebra have an inverse relation, which they can be mapped on to by simply substituting the relation type and switching over the argument order. For example, BEFORE(monday, tuesday) is equivalent to AFTER(tuesday, monday). Automatic classification is easier with a smaller number of classes. We can simplify the task of classifying temporal relations by reducing the set of relation types used.

The procedure of removing inverse relations requires the definition of a set of mappings from relations with their complements. Using this, one removes inverse relationship types by changing them to their original form and flipping argument order. We have named this procedure **folding**.

Various relation folding mappings are available. MITRE specifies one (for example, those used by [21]) and there are mappings to the simple SIMULTANEOUS/BEFORE/INCLUDES relations specified by [10]. To be able to accurately reproduce results, one requires a dataset where the set of relation types has been reduced (folded) in the same way.

Although it may at first seem that folding relations in a document will alter the distribution of relationship classes, it must be pointed out that the exact balance between BEFORE and AFTER relations – indeed between any relation and its inverse – is entirely arbitrary and down to the annotator's personal preference. Folding in

[1] Although scale plays a part here; for some events, starting within the same week or even millennium can be considered synchronous, for others, picoseconds can be considered apart. The final choice is left to the annotator, who should interpret discourse accordingly.

fact removes any influence that annotator preference may have and presents data in a uniform manner.

Based on Table 3.1 from [21], MITRE have opted for the following mappings: (an asterisk indicates that the arguments should be reversed as part of the relation type change)

- IAFTER → IBEFORE*
- BEGUN_BY → BEGINS*
- ENDED_BY → ENDS*
- IS_INCLUDED → INCLUDES*
- AFTER → BEFORE*
- IDENTITY → SIMULTANEOUS
- DURING → INCLUDES*
- DURING_INV → INCLUDES

This gives us a smaller set of six relations, from the original fourteen. The mapping suggested by [10], from [13], is reproduced in the same format here:

- AFTER → BEFORE*
- IS_INCLUDED → INCLUDES*
- IDENTITY → SIMULTANEOUS
- DURING → INCLUDES*
- IBEFORE → BEFORE
- IAFTER → BEFORE*
- BEGINS → INCLUDES*
- ENDS → INCLUDES*
- BEGUN_BY → INCLUDES
- ENDED_BY → INCLUDES

There has been ambiguity over how best to fold DURING relations. After some discussion [22], the TimeML DURING relation can be said to specify a relation between two proper intervals that share the same start and endpoints (cf. "for the duration of") and that DURING is formally equivalent to SIMULTANEOUS; as SIMULTANEOUS is

Table 3.4 Relation folding mappings used in this book

Original relation	Folded to
AFTER	BEFORE*
IS_INCLUDED	INCLUDES*
IAFTER	IBEFORE*
BEGUN_BY	BEGINS*
ENDED_BY	ENDS*
DURING_INV	SIMULTANEOUS
DURING	SIMULTANEOUS
IDENTITY	SIMULTANEOUS

the inverse of itself, nothing unusual need be done for DURING_INV, which resolves to the same type. After this clarification, the fold used in experiments detailed by the rest of this document is shown in Table 3.4.

The effect that folding has on the distribution of link types in the TimeBank corpus can be observed by comparing Tables 3.5 and 3.6.

Table 3.5 Distribution of TLINK relation types in TimeBank 1.2

Relationship type	Count	Percentage (%)
AFTER	897	14.0
BEFORE	1408	21.9
BEGINS	61	1.0
BEGUN_BY	70	1.1
DURING	302	4.7
DURING_INV	1	0.0
ENDED_BY	177	2.8
ENDS	76	1.2
IAFTER	39	0.6
IBEFORE	34	0.5
IDENTITY	743	11.6
INCLUDES	582	9.1
IS_INCLUDED	1357	21.1
SIMULTANEOUS	671	10.5
Total	6418	

Table 3.6 Distribution of relation types over TimeBank 1.2, as per Table 3.5 and folded using the mappings in Table 3.4

Relationship type	Unclosed		Closed	
	Count	Percentage (%)	Count	Percentage (%)
BEFORE	2305	35.9	22033	73.2
BEGINS	131	2.0	226	0.8
ENDS	253	3.9	479	1.6
IBEFORE	73	1.1	169	0.6
INCLUDES	1939	30.2	4368	14.5
SIMULTANEOUS	1717	26.8	2822	9.4
Total	6418		30097	

3.3.1.1 Problems with Folding

While folding reduces the number of possible relation classes and increases the amount of training data available in each class, it introduces some system implementation issues. In controlled evaluation exercises, it is possible to reverse the order of arguments in the evaluation set such that the set only contains relations that the classifier has seen before from folded training data. However, this is not possible in cases where the relation type is never known. One does not have control over the argument order of unlabelled examples that are to labeled. If for example we have removed all AFTER relations from our training data by swapping their arguments and changing the relation to BEFORE, when faced with the previously-unseen relation of (e.g.) "C AFTER D", the classifier will not be able to assign the correct label. One solution is to attempt to classify the intervals twice – A rel B as well as B rel A – and use classifier confidence or the addition of an "unknown" relation type to signify which of the reduced label set should be applied with which arrangement.

Another approach for building applications that can cope with non-synthetic data is as follows. Maintain the normal set of relations and increase training data size by using folding to create a new training instance (instead of folding to alter a training instance) and add that to the set. That is, if we have a training example "A AFTER B", we automatically add an example of "B BEFORE A" and leave both examples in the training set. This technique can be called relation **doubling**. When performing doubling in this manner, it is even more important to partition training and testing data at document and not example level.

In summary: classifiers trained on folded data may not be able to cope with real-world data; classifiers learning from data created by doubling do not have such a disadvantage; folding works by simplifying the training data; doubling works by increasing its volume.

For the sake of comparability, the work in this book is uses training data with folded relations. Investigation of temporal relation doubling as a replacement for temporal relation folding is left for future work.

3.3.2 Temporal Closure

Humans tend to first classify the links where they find the type most obvious, deprioritising other more tenuous or remote links [23]. Thus, out of all possible links between each event and temporal expression, usually only a subset of links are classified by a human annotator. It is possible, however, to determine a canonical version of the temporal structure of a document.

Smaller datasets are problematic for automated approaches to relation typing because they may not contain sufficient information to form generalisations about relations. Further, temporally annotating documents in order to enlarge datasets is a complex and costly procedure. Therefore, any automated aids to increasing the amount of temporal relations annotated are welcome. Fortunately, it is usually possi-

ble to automatically perform some inference over an incomplete annotation, labelling extra edges with relations and thus reducing data sparsity. One may use a temporal algebra to infer relationship types.

Let times and events be nodes on a temporal graph and edges in the graph represent relations between them. Given a partially connected temporal graph (for example, a human temporal annotation of a document), one can iteratively label previously unlabelled edges using an algebra's inference rules. When no more unlabelled edges can be labelled, the resulting graph represents the **temporal closure**. This graph explicitly conveys the maximum amount of information that one is able to deduce from a partial annotation. Once the maximum number of interval pairs have been linked in this manner, we are said to have computed the **temporal closure** of a document. For an example, see Fig. 3.1. Graph-based representations lead to sophisticated reasoning [20] and evaluation measures (Sect. 3.4.4.3).

There is often more than one way of temporally annotating a document's temporal structure. Because there is often more than one way to annotate a document that can be computed to the same temporal closure, when comparing documents, the closure is used rather than the original annotation. Closure also provides extra training examples for supervised learning, which has been explored by many authors, particularly investigated by [24] (see Sect. 3.4.1). We fully investigate comparison of temporal annotations in Sect. 3.4.4.

3.3.3 Open Temporal Relation Annotation Problems

Within temporal relation annotation, there remain open problems in a number of areas. This book contributes towards the solution of one – temporal relation typing. Others are detailed here.

Temporal Relation Identification

This is the task of determining which pairs of events or timexes should be linked. While one may link almost every time and event annotation in a document by means of inference (perhaps through closure), is this the best option? Adding structure to the relation identification task often leaves out some links that are otherwise clear to readers. For example, the TempEval exercises focus on intra-sentence links between the head event and other events, and then on head events between adjacent sentences – but this says nothing about the relation between non-head events in the same sentence. Determining a definition of what constitutes a temporal relation and then finding these in text remain open.

Modality

The majority of research has focused on links between events and times in the same modality and in the same frame of reference. Dealing with modals seems important; they occur frequently, and indeed there is a strong argument that the future tense is

entirely modal. The problem of temporal annotation between non-concrete modalities is open.

Annotation Completeness

How do we know that we've finished annotating? Even given oracles for event annotation, timex annotation, and temporal relation identification and typing, there exists no firm description of what constitutes a complete annotation. Is it when every event and timex is connected? Is it just when those links based upon explicit temporal words and inflections in the text have been annotated? Neither TimeML nor other temporal relation schemas tackle the problem of annotation completeness. As temporal relation annotation in particular is a difficult and time-consuming task, it would be very helpful to establish at least recommended minimum and maximum bounds for relation annotation.

For a really good guide to annotation in general, I recommend "Natural language annotation for machine learning" [25].

3.4 Automatic Temporal Relation Typing

Over the past decade or so, there have been many machine learning approaches to temporal relation typing – the task of determining the relative order (or relation type) between two temporal intervals (which are times or events). Most of these approaches have focused on using a set of relations derived from the 13 labels proposed by Allen (Table 3.1) or a reduced set thereof (e.g. TempEval relations, Table 3.3). The most commonly used datasets are TimeBank and TempEval-2 (Section A.2).

Generally, earlier relation typing systems are accurate in around 60 % of cases and more recent systems reach about 70 % accuracy. This level is only ever exceeded in cases where a subset of all temporal links is examined; never for the general problem.

This chapter describing related work first summarises some concepts particularly useful to temporal relation typing (Sect. 3.4). After this, a set of previous approaches are described, in terms of their dataset, features and performance (Sect. 3.5). The progress in the field so far is then summarised and an analysis presented (Sect. 3.5.6).

3.4.1 Closure for Training Data

In order to provide extra training data, temporal closure [26] can be performed over human-annotated data. This provides a varying number of additional examples, depending on the completeness of the initial annotation (perhaps symptomatic of the lack of a formal definition describing how much should be annotated) and also the text itself.[2]

[2]Examined in greater detail in Sect. 3.3.2.

3.4.2 Global Constraints

In linked groups, temporal relations co-constrain. For example, given:

Example 7

A BEFORE B
B BEFORE C

The set of valid types for an A–C relation is constrained. It is important that automatic labellers take this knowledge into account. The production of an overall inconsistent annotation is a simple thing to check for. In all but the simplest of documents, global co-constraint violates the independence of training examples. In order to preserve separation between training and test data, [24] propose only allowing document-level splits in data.

3.4.2.1 Event Sequence Resources

As we annotate text, it becomes possible to build some discourse-independent record of common event relations. This is essentially a restricted model of world knowledge. For example, we might often see that *travel* happens before *arrive*, or that *sunrise* is included in *the day*. Such records could be used to aid future annotation of unlabelled temporal relation data.

VerbOcean

One such resource that specifies a simple relation between token pairs is VerbOcean [27, 28]. The data comes from mining Google results using templates [29] and then establishing mutual information between mined verb pairs. Different relation types each have their own set of templates. The relations that are useful in temporal information extraction are `[happens-before]` and `[can-result-in]`, reflecting causation and enablement.

Narrative Chains

Chambers and Jurafsky [30] suggest a way of building event chains. These look for common actors in events (either as subject or object) and catalogue the events that the actor participates in. Actors do not need to be people in this context. Event chains are provided in a number of different story types. An example is given where a criminal robs, and then is arrested, and is tried; this sees the "criminal" actor fulfil multiple roles. When a particular chain of events can be seen to occur in the same sequence (with similar actors) over many documents, we can have higher confidence in its accuracy. While this work does not suggest any kind of temporal ordering, it is easy to see how one can build catalogues of temporally sequential stories, which may later be of use when ordering events.

3.4.3 Task Description

The task of determining which times/events to relate is "temporal relation identification". The task of determining the type of relation that holds between a given timex or event pair is "temporal relation typing". This chapter concerns the temporal relation typing task: that is, of assigning on of a set of relation types to a given interval pair, where an interval may be an event or timex.

Consider the sentence in Example 8.

Example 8 The president's son $\underline{\text{met}}_e$ with Sununu $\underline{\text{last week}}_t$.

It contains an event e and timex t. We are told by an external source, e.g. our annotators, that has already performed temporal relation identification, that e and t are temporally related. The task at hand is to choose a relation type from a set of options that best describes the temporal relation between e and t. A list of these options in TimeML is in Table 3.2.

In this scenario, the *met e* seems to occur in its entirety at some time between the beginning and end of *last week t*. So, the suitable relation type is inverse inclusion; that is to say, *e* IS_INCLUDED *t*. Or, the other way round, *last week* INCLUDES *met*.

3.4.4 Evaluation

In many tasks related to temporal processing of text, there is a need to compare annotations. One may want to compare two human annotations, or measure how favourably an automatic annotation compares to an existing gold standard. Developing an automated temporal information extraction tool in any kind of scientific way requires formal evaluation. Comparing two human annotations will give values for inter-annotator agreement (used as a rough cap for automatic annotation performance) and the ability to evaluate automatic systems is essential.

Human annotation of temporal relations is difficult [10, 31]. This is sometimes caused by a lack of context during annotation. For example, some systems show only two event sentences, omitting surrounding discourse which may contain clues [32]. Humans, for example, have trouble distinguishing some relations such as IS_INCLUDED and DURING [33]. The temporal relation annotation task is complex enough to have a large number of idiosyncratic difficulties, which we can only identify through annotation comparison.

In the rest of this section, we introduce general issues with temporal relation evaluation and then discuss the application of traditional precision and recall measures to this task, as well as two graph-based methods for comparing temporally-annotated documents.

3.4.4.1 General Issues

Temporal relation annotation evaluation involves the assessment of relation type assignments between an agreed set of nodes. Because of the complex nature of the interactions between relations that share nodes, the following issues need to be taken into consideration when evaluating temporal relation typings.

Firstly, with most relation sets there is more than one way of annotating a single relation between two events or times. One may say "A BEFORE B" or "B AFTER A", both describing the same temporal relation between A and B.

Secondly, the transitive, commutative and co-constraining nature of temporal relations in a network mean that there are many different ways of representing the same information [10, 26, 34] in the form of a temporal closure. As a result, missing links are not always a problem, as long as the information required to infer them is present somewhere in a document. As a general approach, one should only evaluate over the closure of a document's annotation.

Finally, when evaluating it is important to take account of which document an instance of a relation comes from. Mutual co-constraint means that relations within a single document or temporal graph are not independent. When partitioning data into training and test sets, one must be careful to split at document level; that is, all links from any document should be in the same set. When performing cross-validation, all of each document's links should be found only in one single fold [24].

3.4.4.2 Precision and Recall

Annotations can be compared in different ways. When evaluating automated TIMEX identification or relation classification against a gold standard, we can measure precision and recall. For example, one can use these metrics to describe the amount of TLINKs correctly found in a candidate annotation versus a reference annotation. TimeBank is often used as a gold standard for training and evaluation of systems using TimeML. Evaluating TIMEX normalisation needs a different measure, as there are varying degrees of correctness available; one has to take granularity into account, as well as potentially overlapping answer intervals, which should not automatically be granted zero score.

Sometimes important links will be missed by annotators; sometimes multiple unclosed annotations of the same closed graph can differ. The latter can be compensated for by only comparing closures; in fact, precision and recall should only be measured between closed graphs, otherwise there is misleading ambiguity between different representations of the same information. Measuring the presence of relations only affects recall; unlabelled edges are equivalent to missing information, as opposed to incorrect information.

3.4.4.3 Graph-Based Evaluation

While precision and recall provide an indication of the closeness of two annotations, they are imperfect in the context of temporal annotation. Flaws exist in relation type matching and evaluating interval boundary point assignment. For relations, some temporal link types are more closely related than others. If we guess INCLUDES when the real answer is ENDED_BY, we have done much better than if we guess BEFORE. For intervals, working at interval level requires both endpoints to be correct before awarding a full entity match. However, it is rational to issue a partial reward if one endpoint has been found correctly, when compared to cases where neither are correct. Precision and recall based systems cannot directly cater for these features of these problems. This section discusses a graph-based evaluation metric that attempts to address these issues.

As mentioned in the chapter introduction above, a discourse's temporal information can be imagined as a graph (see Sect. 3.2.2). Temporal closure of the graph can be computed, leading to a more consistent representation of the annotated data [3, 10, 13]. It is possible to measure agreement between graphs [32].

Not all relations have the same importance; some entail more information – some may lie on something akin to a critical path [35], and conversely some may only be dead ends that do not affect the rest of the graph. Resolving certain relations provides more information than others. Thus, a metric that rewards the labelling of the most important edges is required.

One can use a graph algebra to build a metric for graph similarity. One method of achieving this, proposed by Tannier and Muller [34], involves the following steps:

- Graphs between events are converted into graphs between points
- Each event is split into a beginning and end point
- Only equality (=) and precedence (<) relations are needed
- Two nodes linked by equality relations are merged

This produces an acyclic directed graph, of arcs which represent precedence relations, and nodes that represent collections of temporally simultaneous points. An edge between time points x and y implies that x is equal to or less than y. The transitive reduction of a directed acyclic graph, which is unique, is calculated. After this, Allen relations are converted into '=' and '<' (equality and precedence) relations between endpoints. At this point, we have a linear directed graph, with one or more points (each representing an interval start or end point) at each node. From the directed graph, multiple candidate graphs can be compared by the number of manoeuvers required to reach one graph from the other, in a similar fashion to establishing a Levenshtein edit distance [36].

Manoeuvers are of two types. A **split** is where a node is broken and a **merge** is the addition of a point to a node.

The similarity between graphs is measured based on the number of merges and splits required to transform them, over the total number of relations. One can then calculate a revised version of 'temporal' recall and precision, based on features in the graphs. Graph value, representing the size and complexity of a graph, is key to

these measures. It is also possible to evaluate graphs that include temporal relations of the form 'before or equal" ('after or equal" is reduced to this form by reversing arguments). Half-splits and half-merges can be introduced, with an initial weighting of 0.5 for the move, where a half-split would be the removal of a point related with such a disjunction.

To see how useful this evaluation metric is, its authors used it to examine graphs where selections of temporal relations had been removed from minimal graphs and a linear decrease in the standard recall measure was observed (as expected). However, while recall harshly penalises graphs that lack some critical information, this metric still rewards the remaining partial information, leading to a convex graph curve, which can be seen in Fig. 16 of [34]. Thus, this measure provides an intuitive metric for temporal annotation comparison which offers partial rewards for partially correct information, unlike precision and recall measures.

Although an improvement upon earlier metrics, graph-based evaluation is used little in the literature and so experiments measured using can be difficult to compare to previous work; e.g. [37].

3.4.4.4 TempEval

The TempEval semantic annotation evaluation exercises are shared tasks focusing primarily on temporal relation annotation. They have also served to advance the state of the art in temporal annotation [38]. TempEval and TempEval-2 both use a simplified set of relations and a purpose-created corpus. Systems in TempEval-2 [39] showed some incremental relation typing performance improvements over the previous exercise. While the first TempEval focused on the temporal relation typing task, TempEval-2 added event and timex annotation, and TempEval-3 [40] also required participants to perform temporal relation identification. These three establishing evaluation challenges led to us seeing a proliferation of temporal evaluation challenges; in 2015 we saw not one but four different temporal shared challenges at SemEval, covering cross-document coreference and ordering, question answering, clinical data, and document data [41–44]. Clearly temporal semantic annotation is an area full of tough and fundamental challenges.

TempEval has generally contributed extra data and served to advance the state of the art, not only by stimulating research as many different sites contribute systems but also by providing empirical, comparable results for many different approaches to temporal annotation.

3.5 Prior Relation Annotation Approaches

This section presents an overview of automatic temporal relation typing efforts. It aims to be comprehensive, especially to include work done after the introduction of TimeML. It is broken into the discussion of machine learning-based systems,

rule-based systems and hybrid systems. Several techniques for boosting training data size and feature effectiveness are discussed. Finally, an analysis is presented in which successful parts of an approach are identified and future work is outlined.

3.5.1 Feature and Classifier Engineering

Many approaches have relied on using example relations to train a classifier, i.e. are supervised learning approaches. These relations are represented as a vector of features. It is critical to select the right features and classifier, and these have been topics of many prior approaches.

Machine learning approaches do not require an intimate and accurate human understanding of all linguistic relations within a document. Rather, a classifier learns rules or models from training data and uses these to attempt to predict the label of future relations given their feature vector representation.

Classifier performance generally improves as more training data becomes available. This has the benefit of being able to directly boost performance through data collection. However, insufficient training data can lead to poor performance, and in the context of temporal annotation, collecting more data is expensive. In the case of temporal information extraction, relatively small amounts of ground truth data are available.

With linguistic datasets, it is important to choose a classifier that can resist some noise in its training data. Natural language is robust and many utterances can be understood despite some minor mangling. Further, the diverse range of words that may be used in any situation are prone to inducing overfitting if not handled correctly. We shall see this later, in for example Sect. 5.6.3.5.

One of the earliest approaches [45], shortly after the release of TimeBank 1.1 (which included timex, event and relation annotations), attempted to both determine which intervals to link (the relation identification task) and then also to determine the nature of the TimeML relation between detected pairs (the relation typing task). It used an RRM classifier [46] to jointly detect and label TLINKs based on features derived from a finite state parser. These were based on the gold-standard event and timex annotations in that corpus. Only event-timex links were considered. A proximity threshold for intervals classified as being temporally linked or not was set. This proximity threshold was varied in an attempt to discover its impact on the complexity of the task. The baseline for pairing was that only if an event and timex were the closest of their kind to each other would a link be said to exist, and the baseline for typing was most-common-class (IS_INCLUDED). Features are based on part-of-speech tags, word shapes, syntactic chunk information and n-grams.

Only looking for TLINK argument pairs within 4 tokens provided the strongest results at the pairing task (F-measure 81.8). When the authors have to both find the TLINK and then assign a relationship type (a harder task than we address in this book), the F-measure dropped to 58.8. This indicates a typing accuracy of around 70 % in this small subset of TLINKs. Adding FS grammar information (see also

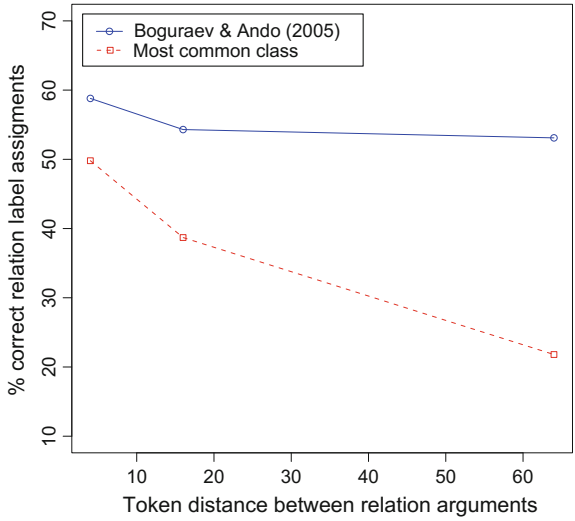

Fig. 3.2 TLINK relation type assignment difficulty increases with the distance between link arguments

Sect. 2.3.3.1) to the feature set consistently provides a small absolute performance boost (0.7–1.8 %). They found that automatic detection and typing was easier for relations between intervals the closer that they were in discourse, reaching 58.8 % accuracy on the joint TLINK-finding/relationship assignment task for interval pairs within four tokens of each other (which accounts for 12 % of TimeBank's relations). This accuracy decreased with larger token window sizes (see Fig. 3.2, which is derived from data tables in their paper). Considering EVENT/TIMEX3 pairings in the largest window size – 64 tokens – yields a low baseline performance of 21.8 %; the classifier improves on this to reach 53.1 % at this joint relation identification/typing task.

It is possible to determine the performance of [45]'s joint relation identification/ approach at just the relation typing task. Dividing joint pairing/typing performance by typing performance gives the typing accuracy over correctly identified relations. In this case, for 4-, 16- and 64-token windows respectively, TLINK typing using the features above including FS grammar information reached 71.9, 71.0 and 71.0 % accuracy respectively. These figures apply to event-timex links between intervals that appear relatively close to each other in discourse.

As part of TempEval 2007, [47] experimented with a range of classifiers and the basic event/timex attributes as features, attempting to gather information on which attributes were helpful in relation typing. Among other things, they found that tense and aspect features were of less use in event-timex relation typing than in event-event, and that SVM and K* classifiers performed best.

After the release of TimeBank v1.2, upon which the majority of recent temporal relation extraction work is based, [21] proposed a supervised learning approach to event-event and event-time relation typing, using the interval pairings specified in the corpus. This was refined and presented later [24] as an approach that provides a useful baseline for other supervised approaches, as it relied only upon information

annotated with TimeML (e.g. no n-gram or syntactic features). The features used for each link were the text and TimeML element attributes of the intervals comprising the link, as well as a few simple Boolean features describing whether or not the tenses and aspects of both participants in an event-event relation were the same. The authors experimented with using temporal closure to increase the number of relations available (see Sect. 3.3.2).

The corpus used is a merging of a custom version of TimeBank [48] (v1.2a – not publicly available) and the Aquaint TimeML Corpus (ATC) [49]. Applying a maximum entropy classifier (from Carafe[3]) reaches an accuracy of 82.5 % when classifying event-to-time relations, better than the most-common-class baseline of 65.5 % (this class is the INCLUDES relation). Event-event relations were labelled with 59.7 % accuracy, which improved on the most-common-class baseline of 51.7 % (BEFORE). Other classifiers – namely SVM and naïve Bayes – performed similarly. As for using data from temporal closures of the annotations in the source corpus, event-time typing was better than baseline but overall worse (71.2 % accuracy, 51.3 % baseline) but event-event typing did worse than most-common-class baseline (51.1 % accuracy, 54.1 % baseline). Generally, classifiers trained on unclosed data performed better when predicting labels for TLINKs from unclosed data than did classifiers trained on closed data (at predicting TLINKs from closed data). This suggests that simply generating extra feature instances via temporal closure of source data data is not an effective method for learning better classifiers.

Later approaches have adopted the method used by [24] – that is, using a combined TimeBank/AQUAINT corpus plus the TimeML element attributes as features. Using support vector machines, [50] achieved performance gains in TimeML temporal relation typing using syntactic tree kernels. Their approach reached 80.04 % accuracy on event-time links in ATC using a polynomial composite kernel (compared to 82.47 % from [24]) and 67.03 % for event-event relations on the same (compared to 70.4 % from [51], detailed below).

Vasilakopoulos [52] use a K* approach to temporal relation typing. They determine the most useful features for the typing task and discard the least useful, as well as experimenting with new semantic features. This leads to strong performance on the earlier TimeBank 1.1 corpus.

3.5.2 Rule Engineering

As opposed to supervised machine-learning approaches, some approaches to automatic temporal relation typing use a human-engineered set of rules to determine how to assign a relation label. These rules are typically based on information about the relation and its arguments. These approaches can be simple and intuitive and quickly achieve above-baseline performance with a minimal ruleset. However, to reach com-

[3] Available at http://sourceforge.net/projects/carafe/.

petitive accuracy levels, the rule set generally becomes more complex and harder to understand.

Rule based approaches tend to be more fragile than generic learned approaches. Extrapolation can be a particularly difficult task, which can occur when coping with unseen data that does not match patterns previously seen. Further, performance is not dependant on the amount of training data, but instead the quality of the rule set. Therefore, one cannot directly turn extra data into better accuracy.

That said, there are still some rule-based approaches that have met with success. Initial work on the relation typing task was conducted by [53], using a rule-based technique to anchor events to times. This rule-based technique draws on principles from Reichenbach's model of tense and aspect [54]. They achieve an 84.6 % accuracy, though the work is hard to compare to later approaches based on TimeML because the relation set is simplified and the event and time definitions are not the same.

It is possible to add rules to a system which support incorrect decisions in some cases. Such rules will damage performance. However, including only high-performance rules becomes increasingly difficult as more rules are added to a system, and can constrain the scope of new rules to only cover a few cases. Kolya et al. [55] describe a rule-based approach that includes rules which have known contradictions in the training dataset. This approach has intentionally capped its maximum performance. Despite this, is it still able to achieve reasonable accuracy on its evaluation set.

The sentiment that neither rule-based nor statistical methods alone can satisfactorily solve a qualitatively described real-world problem is not a new one [56]. Hybrid approaches can overcome problems with both rule-based and machine learning-based options. Rule based systems have problems with rigidity and with their high construction cost; machine learning systems can quickly make inferences over data, but rely on having both accurate data and enough data. With a hybrid system one can incorporate rules to quickly achieve a base performance level and a machine learning component can "weight" rules to avoiding some of the fragility of complex rule bases. Further, one can quickly and simply prototype a machine learning system and then provide expert knowledge in the form of rules, allowing a rapid way of building new information into an automatic labeller. As a result, rule engineering has been used in combination with machine learning by many approaches to the relation typing task.

Kolya et al. [57] augment a CRF-based event-time relation typing system with a set of hand-crafted rules that encode observations about the dataset, leading to strong performance for event-event and event-time relation typing. In later work they take a similar approach [58], using event head information to achieve reasonable TempEval-2 scores.

3.5.3 Syntactic and Semantic Information

Syntax is often used alongside lexemes to convey the meaning of an utterance. It is therefore reasonable to investigate the effect of syntactic and semantic information on the temporal relation typing task, as many prior approaches have.

Following [24, 59] add features describing temporal signals, syntactic and semantic roles, and perform reasoning about the context events and timexs appear in to see if they are within one context. They participated in the TempEval challenge, which was not based upon TimeBank but a smaller dataset with a smaller set of potential relation types. They obtain 0.55 accuracy on TempEval's E-E relation typing task using an SVM, which matched the best performance in this task and beat the baseline of 0.47.

During TempEval-1, top performance at event-event relation typing was given by a rule-based system, XRCE-T [60], which relied on deep parsing using a custom parser, XIP. This performance was later matched by a system based on machine learning and notably more complex information sources [61].

Syntactic relations can also play a role in determining temporal relation types. For example, Bethard et al. [62] combine event and syntax features to train an SVM kernel that reaches 89.2 % accuracy on a selected set of event-event relations in TimeBank using a simplified set of three temporal relations. Their feature set includes values that depend upon particular types of syntactic relation between the arguments of a temporal link. Their dataset is constrained to only those event pairs where one event syntactically dominates another.

From TempEval [32], it was observed that performance on tasks that required relation identification between two events or times within the body of the document was low (as opposed to links to the document creation timestamp). One could hypothesise from this that the syntactic structures that connect this pair of lexicalised intervals have some impact on their temporal relation type. To test this hypothesis, [33] created a custom corpus of verb-clause event pairs, using TANGO (see Section A.3.1) and the TimeBank guidelines, with additional annotation rules covering modal/conditional events, aspectual links and permissive verbs (such as 'allow', 'permit' and 'require'). After this, relation identification was modelled using two sets of features; a linguistic set based on event verbs, including things such as tense and aspect and another set based on connecting words (such as signals). This connecting word set included some string features, as well as information about syntactic path and two features based on bags of interconnecting words. Top features were mostly related to target-path (syntactic node path from a clause to its head) or to the subordinated event. Increased word-distance between events decreased relation typing performance, just as was the case in [45].

Cheng et al. [63] use dependency parsing to generate features for relation typing, coupled with a sequence labelling model for events. They assume that, since time is linear, events occur in order, and therefore the events in a document can be treated as a sequence. This leads to an interesting HMM model for inter-event relation typing. Similarly, UzZamana and Allen [64] use a rich, in-depth parser to support their

features for a Markov logic network when typing temporal relations. This lead to the best score for event-time labelling in TempEval-2.

As part of a syntacto-semantic approach to temporal information extraction, including timex and event annotation, [65] built on their earlier approach [66] and used syntactic analysis for the event-time relation typing task, also post-correcting classifier output using a system of hand-crafter rules. The approach placed special focus on clause graphs, and achieved moderate success at event-time relation typing.

Ha [67] used a set of lexico-syntactic features for events and times to learn a Markov logic network as a model for temporal relations with a given document. The approach draws additional information from VerbOcean and WordNet. This intuitive approach performs well at event annotation, but extra analysis is required to improve relation typing performance.

Semantic roles have been found to play a useful role in both interval (i.e. event and timex) annotation and temporal relation typing [68]. The concepts are further explored in [69], finding that tense information can be misleading, but still achieving a performance increase over TempEval-2 systems.

3.5.4 Linguistic Context

Some prior approaches rely on discourse information not annotated with TimeML, which typically only applies to a small proportion of tokens in any given text. Looking at the document as a whole, and the linguistic context in which events and timexes lie, may lead to improved relation typing performance (Table 3.7).

VerbOcean is a resource detailing semantic relations between verbs, mined from large corpora. One of these relation types is temporal: *"happens-before"*. Ref. [21]'s system includes experiments which perform VerbOcean (Sect. 3.4.2.1) and GTag[4] rule lookups and use the results as features for machine learning. The data sparsity of VerbOcean leaves it contributing only very slightly to results, to the point where it is hard to tell if performance increases are statistically significant. Out of 24 instances where VerbOcean matches could be made, 19 correctly suggested the final relationship type; 5 incorrect results were found.

The best results are when the scope of TLINKs studied is heavily constrained and situation-specific features used [62, 70]. However, when the features that help in these specific situations are applied generally, they lead to a performance drop in typing of other TLINKs. This suggests that it may be best to apply different typing techniques to particular subsets of TLINKs, instead of trying a "one size fits all" approach.

[4] "GTag takes a document with TimeML tags, along with syntactic information from part-of-speech tagging and chunking from Carafe and then uses 187 syntactic and lexical rules to infer and label TLINKs between tagged events and other tagged events or times." [21].

Of the mechanisms that play a part in conveying temporal relational information, one that has been under-investigated is the use of expressions, typically adverbials or conjunctions, which overtly signal temporal relations – words or phrases such as *after*, *during* and *as soon as*. Very few of the teams participating in the recent TempEval challenges [38–40] exploited these words as features in their automated temporal relation classification systems. Certainly no detailed study of these words and their potential contribution to the task of temporal relation detection has been carried out to date; this is the subject of Chap. 5.

As part of a TempEval system, [71] attempted to find temporal "signal" words – those word which act in a temporal sense to make explicit the nature of a temporal relation, such as *"simultaneously"* – and use these to augment a MaxEnt-based relation labelling system. The approach yielded a mild improvement. Further investigation was given into the impact these signal words can have on the relation typing task [70], showing them to be capable of giving an error reduction of over 50 % for TLINKs that are associated with one. Temporal signals are the focus of a later chapter in this these (Chap. 5).

This has continued through to recent TempEval tasks, such as Clinical TempEval [43, 72], which implements narrative containers as a temporal structuring device [73]. These are defined as "the default interval containing the events being discussed", and implemented in order to increase the informativeness of temporal annotation [74].

Finally, [75] experiment with the addition of event participant and event coreference features, using an SVM to label relations. This achieves a modest performance level on the event-event relation typing task.

3.5.5 Global Constraint Satisfaction

As temporal relations co-constrain, it can be said that the type of one relation may have a bearing on the types of other relations between which an endpoint is shared. Therefore, considering these global relation type constraints is important to achieving a correct overall relation typing solution, and may lead to improvements in the assignment of individual label types.

Chambers and Jurafsky [51] manually add links to TimeBank v1.2 in cases where events subordinate other events in the same clause (as per [62]) and links between calendar times. They then perform closure and folding over this extended dataset in order to generate extra training examples for an SVM classifier. The output from this classifier is then processed through a model that ensures that temporal relations are globally consistent, correcting relation labels where necessary. No overall accuracy is gained, though after the problem is reduced to just before/after relations, this post-classifier-typing correction yields a 3.6 % accuracy improvement.

Later, [61] use a Markov logic network to model constraints and obtain top accuracy on TempEval's relation typing task. They find that using Markov logic allows better capture of non-absolute rules between relation pairs and that a model need only

be built once instead of per-document, which moves focus onto temporal relations instead of the mechanics of machine learning.

3.5.6 Summary

Although event-time relation typing accuracies can reach as high as 80 % (as in e.g. TempEval), overall temporal relation typing performance has stalled around 70 % accuracy, leaving temporal relation extraction an open research challenge. Applications require higher performance, but it is not available. Current accuracy is too low to support NLP tasks such as question answering [76], forensic analysis [77] or temporal slot filling [78, 79].

From the above, we can see that classifier choice affects relation typing performance, even for different relation argument types. Including data on global temporal constraints, on syntactic structure and on tense modelling can all help. Further, we see that generic approaches obtain quite different performance in different TLINK settings (such as in TempEval).

Hand-engineering and machine learning methods are effective, even when rule bases have built-in failings. Machine learning methods have reached a performance cap. Improving temporal relation typing accuracy becomes increasingly hard and performance appears to have almost levelled off. Extra effort and sophistication in relation typing approaches yield diminishing returns.

3.5.6.1 TimeML Features

Relying on only the TimeML attribute values as features is not sufficient. Machine learning approaches that use this set of features seem unable to break through the 70 % event-event relation type accuracy barrier, even on folded data [80] or after attempts with a sophisticated array of cutting-edge classifier kernels [81, 82]. Even the introduction of some syntactic information such as argument ancestor path distance and is not sufficient to overcome this barrier [50, 83]. Taking care of other information sources, such as global constraints, yields an immediate but small performance increase over the base feature set [51, 61].

Despite almost a decade of work, relation typing accuracies over even 80 % are a rare event. This is suggestive of some greater difficulty that has not yet been identified. It is possible that there is simply not enough training data, and that generating more through closure is somehow not sufficient (this does not yield performance improvements); this is investigated in Sect. 3.6.1. It could also be the case that TimeML is structurally insufficient somehow, e.g. the markup's attributes and values may be insufficient for capturing all the information required to type a temporal relation. Also, as the highest performance levels are seen on subsets of links from a whole

corpus, there may be merit in subsetting relations somehow and working to understand each group. Finally, other problems could arise from the task being insufficiently well-defined, which may manifest in poor inter-annotator agreement. We discuss how well-defined the task is in the rest of this section and relation subsetting in the next chapter.

3.5.6.2 Task Definition Issues

Regarding the definition of the task, there is some data available to describe how well it is understood. In temporal link annotation, separate inter-annotator agreement (IAA) figures are given for relation identification and relation typing. For TimeBank 1.2, relation identification IAA (i.e. the extent to which annotators agreed which pairs of intervals should be related) was low – around 0.55 – though is attributable to the fact that a single temporal relation structure of a document can be described in multiple ways, all equivalent after closure. Unfortunately, IAA figures are not given post-closure, but only pre-closure, and so this 0.55 is a minimum.

Critically, relation type annotation agreement is 0.77 – not absurdly low but below the recommended 0.90 [84]. State-of-the-art in performance overall performance is around 72 % accuracy, which is below IAA, though current performances are nearer to IAA than they are to baseline performance.

There are multiple relationship sets available, and the Allen set used by TimeBank has faced some criticism (e.g. [16]). TempEval-1 and TempEval-2 involved the annotation of data with an alternative (and simpler) relation set. IAA these annotation tasks may be compared to that from TimeBank's to see the impact of reducing the relationship set's complexity on annotator agreement. For TempEval-1, event-time IAA was 0.72 and event-event IAA 0.65. Agreement scores are not readily available for TempEval-2.

When measuring the task difficulty using IAA, it is important to note that not all annotator disagreements are equal. Some relations are temporally equivalent. Disagreeing between SIMULTANEOUS and IDENTITY reduces IAA but the final annotations describe events happening at similar times. Other relations are very close. For example, IBEFORE and BEFORE describe almost the same relationship and temporal ordering. Many relationships place intervals in arrangements where one interval bound is in the same place, but the other is not. When one compares A INCLUDES B with A ENDS B, the start point of interval A is positioned between the start and end points of B – it is only the arrangement of A's end point that these relations disagree upon. TimeML's use of an interval algebra means that the position of both points of both intervals in a relation must be specified. Therefore, it only takes the start or end bound of either of the intervals to be slightly vague for the relationship type to become ambiguous to annotators, fostering annotation disagreement (for details, see the TimeBank corpus notes, e.g. Table A.1).

Table 3.7 Prior work on automatic temporal relation classification. As event-event (E-E) linking is generally a harder task than event-time (E-T) linking, results are in ascending order of event-event relation typing performance. In the case of TempEval results, event-event linking is measured as performance at linking main events in consequent sentences and event-time link is matched to the task of linking events and timexes in the same sentence. Therefore, for TempEval-1, the last two columns correspond to tasks C and A respectively. For TempEval-2, the last two columns correspond to tasks E and C respectively. All TempEval results are for "strict" evaluation

System	Notes	Method	E-E	E-T
Lapata 2006 [85]	BLLIP corpus	Decision tree	70.7	
Gaizauskas 2006 [86]	Clinical corpus	Rule-based	65	
Bramsen 2006 [87]	Medical discharge summaries	Graph based	78.3	
TempEval-1 corpus				
Baseline	*Most common class*		*47*	*57*
Cheng 2007 [63]	Uses dependency parsing	HMM SVM	49	61
Hepple 2007 [47]	Includes text order features	SVM/K*	54	59
Bethard 2007a [88]	Uses syntactic tree features	SVM	54	61
Marsic 2011 [65]		Rule-based		65
Kolya 2011 [55]		CRF + rules		75.9
Puscasu 2011 [66]	Syntactico-semantic features	rule-based	54	**80**
Min 2007 [59]	Focus on rules for marginal cases	SVM	55	58
Kolya 2010 [57]		CRF	55.1	73.8
Hagege 2007 [60]	Based on XIP deep parse data	rule-based	**57**	34
Yoshikawa 2009 [61]	Models global TLINK constraints	MLN	**57**	65
Bethard 2007b [33]	Same-sentence links only	SVM + rules	89.2	
Costa 2013 [89]	Tense, aspect and interval relation rules	Various WEKA	77.9	68.0
TempEval-2 corpus				
Baseline	*Most common class*		*48.63*	*55.07*
Derczynski 2010a [71]	Includes signal information	MaxEnt + rules	45	63
Ha 2010 [67]	Lexico-syntactic feat. + VerbOcean	MLN	51	63
Llorens 2010 [68]	Includes semantic features	CRF	55	55
Kolya 2010 [58]	Includes event head information	CRF	56	63
UzZaman 2010 [64]	Based on TRIPS parse data	MLN	58	65
Hovy 2012 [90]	Tree kernel with bags of [words, PoS tags]	SVM	–	64.5
Laokulrat 2014 [91]	Timegraphs, pairwise entity similarity	Stacked learning	**59.7**	**65.9**

(continued)

Table 3.7 (continued)

System	Notes	Method	E-E	E-T
TimeBank 1.1 corpus				
Baseline	*Most common class*		*33.38*	
Boguraev 2005 [45]	Token windows, FS-grammar features	RRM		53.1
Vasilakopoulos 2005 [52]	Not using folded relations	K*	53.14	
Chambers 2007 [80]	Segregates intra-sent. relations	SVM	**67.57**	
TimeBank 1.2 corpus				
Baseline	*Most common class*		*38.35*	*58.4*
Puscasu 2007 [92]	Maps to TempEval relations	rule-based	53	65
Tatu 2008 [75]	With actor and co-ref features	SVM	58.2	
Mirroshandel 2010 [83]	Bootstrapped kernel w/ AAPD	SVM	66.18	
Chambers 2008 [51]	Models global TLINK constraints	SVM + rules	70.4	
Combined TimeBank 1.2 and AQUAINT TimeML corpus				
Baseline	*Most common class*		*51.57*	*65.3*
Mani 2007 [24]	Uses TimeBank 1.2a	MaxEnt	(59.68)	(82.47)
Mirroshandel 2010a [50]	LICT Polynomial kernel	SVM	67.03	**80.04**
Mirroshandel 2010b [83]	Bootstrapped kernel w/ AAPD	SVM	68.07	
Derczynski 2010b [70]	Signalled TLINKs only	MaxEnt	82.19	

3.6 Analysis

So far, we have shown that general temporal relation typing performance is limited to around the 70 % level (and often not far from the baseline), and that the state of the art isn't moving. This section discusses possible causes, and identifies what does seem to work based on prior efforts.

3.6.1 Data Sparsity

There is not enough annotated data to cover all the combinations of values available through TimeML. This means that there is a chance of seeing new sets of data values that do not exist in any prior labeled dataset. TimeBank has about 6 000 TLINK annotations. Each of these constitutes two arguments (each either a timex or event annotation), a relation type and optionally a reference to text supporting the relation type. Aside from the text that they annotate, events have a class attribute (that has one

of seven values), a part-of-speech tag (five choices), a tense (seven choices), an aspect (four choices), and a polarity (two choices) plus cardinality and modality which are free choice (there are 25 values of modality and 15 of cardinality shown in TimeBank). This gives up to 7*5*7*4*2*25*15 = 735 000 possible event configurations (ignoring the free-form lexicalisation of the event). In the simplest case, ignoring event text and text supporting relation types, this makes about $5.4 * 10^{11}$ possible attribute configurations for an event-event temporal relation. The sparseness with which event attribute space and temporal relation attribute are populated by human-annotated corpora means that we are almost certain to encounter previously-unseen combinations of attribute values when attempting the relation typing task on new data. Further, it constrains our ability to make accurate generalisations based on the data that has already been seen.

3.6.2 Moving Beyond the State of the Art

To improve performance in the relation typing task, it is important to understand where the problems are and to determine promising directions for further investigation. Some parts of TempEval-1 have been analysed and there are some trends visible even in our small dataset of temporal typing approaches.

Lee [93] provides an error analysis of TempEval-1. Failures are broken down in terms of relation features, such as relation type, argument PoS and tense. It is found that relations of nominalised events are particularly difficult to predict, as are relations where at least one argument is part of reported speech. Data sparsity is a constant problem, with the less-frequent relation types often failing. This error analysis, while enlightening, does not include any attempt to explain or characterise the harder links or to determine if there is a common difficult set.

As for specific tools, Markov logic networks are likely a useful tool for simply modelling global temporal constraints without placing too much restriction or dependency between individual relation labels. They could also help capture knowledge embodied in successful rule-based approaches while being flexible on the known-imperfect rules.

The problem could also lie with representation. The Clinical TempEval series uses narrative containers instead of interval relations; while comparable machine learning performance can be achieved over this representation [94], the inter-annotator agreement issues still stand. It is a hard task [95]. Empirical evaluation suggests that the more expressive representations are harder for statistical learning [96], though insights into human annotation are certainly needed if we are to develop solid temporally-annotated resources.

It is apparent that no single approach has been able to classify a complete set of links; in fact, usually at least a third are mistyped. It would be prudent to conduct an error analysis, in an attempt to characterise the kind of information that one could use to label mislabelled relations. It may be that there is a consistently mislabelled set

of "difficult" links within the datasets. Examining these may provide insights in to how to improve temporal relation typing accuracy.

3.7 Chapter Summary

This chapter discussed how we may represent temporal orderings between times and events (temporal intervals). It introduced ideas of point-based, interval-based and semi-interval based temporal relations. A literature review is also included, describing historical and modern systems for automatic annotation of temporal relations. The finding is that general-purpose temporal relation annotation systems have hit a performance ceiling at only modest accuracy. Among other tools, the case is made for a failure analysis of current temporal relation labeling systems.

Descriptions of the concept of a temporal relation, were included offering formal definitions, reasoning algebrae and annotation schemas for temporal relations. These foundations were followed by a review of previous work in automatic temporal relation extraction. It has outlined many sets of approaches, drawing upon statistical methods and rule-based methods; using machine learning and human-engineering systems.

As part of the literature review, evidence was presented that current approaches to the temporal relation typing problem are insufficient and more information than available in the TimeML features may be needed. Further, it is noted that the most successful approaches are those that have focused on a subset of temporal relations that have particular properties. This supports our hypothesis that to understand how to temporally order events described in text, we need to draw upon multiple heterogeneous information sources.

The next chapter will conduct an empirical failure analysis of the link typing task, examining particular subsets of temporal relations and how they may be automatically labelled. Along with a baseline method, these are proposed as avenues of investigation for the later parts of this book.

References

1. Setzer, A., Gaizauskas, R.: A pilot study on annotating temporal relations in text. In: Proceedings of the Workshop on Temporal and Spatial Information Processing, vol. 13, pp. 1–8. Association for Computational Linguistics (2001)
2. Setzer, A.: Temporal information in newswire articles: an annotation scheme and corpus study. Ph.D. thesis, The University of Sheffield (2001)
3. Allen, J.: Maintaining knowledge about temporal intervals. Commun. ACM **26**(11), 832–843 (1983)
4. 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)
5. Prior, A.: Tense logic and the logic of earlier and later. Papers on Time and Tense, pp. 116–134 (1968)

6. Bruce, B.: A model for temporal references and its application in a question answering program. Artif. Intell. **3**(1–3), 1–25 (1972)
7. Moens, M., Steedman, M.: Temporal ontology and temporal reference. Comput. Linguist. **14**(2), 15–28 (1988)
8. Goranko, V., Montanari, A., Sciavicco, G.: A road map of interval temporal logics and duration calculi. J. Appl. Nonclassical Log. **14**(1/2), 9–54 (2004)
9. Denis, P., Muller, P.: Comparison of different algebras for inducing the temporal structure of texts. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 250–258. Association for Computational Linguistics (2010)
10. Setzer, A., Gaizauskas, R., Hepple, M.: The role of inference in the temporal annotation and analysis of text. Lang. Res. Eval. **39**(2), 243–265 (2005)
11. McDermott, D.: A temporal logic for reasoning about plans and actions. Cogn. Sci. **6**, 101–155 (1982)
12. Galton, A.: Temporal logic. In: Zalta, E. (ed.) The Stanford Encyclopedia of Philosophy. Stanford University, The Metaphysics Research Lab (2008)
13. Vilain, M., Kautz, H.: Constraint propagation algorithms for temporal reasoning. In: Proceedings of the Fifth National Conference on Artificial Intelligence, pp. 377–382 (1986)
14. Tsang, E.: The consistent labeling problem in temporal reasoning. In: Proceedings of the AAAI Conference, pp. 251–255. AAAI Press (1987)
15. Allen, J.F., Hayes, P.J.: Moments and points in an interval-based temporal logic. Comput. Intell. **5**(3), 225–238 (1989)
16. Freksa, C.: Temporal reasoning based on semi-intervals. Artif. Intell. **54**(1), 199–227 (1992)
17. Kowalski, R., Sergot, M.: A logic-based calculus of events. In: Foundations Of Knowledge Base Management, pp. 23–55. Springer, Heidelberg (1989)
18. Verhagen, M.: Times Between The Lines. Ph.D. thesis, Brandeis University (2004)
19. Derczynski, L., Gaizauskas, R.: Analysing temporally annotated corpora with CAVaT. In: Proceedings of the Language Resources and Evaluation Conference, pp. 398–404 (2010)
20. Denis, P., Muller, P.: Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (2011)
21. Mani, I., Verhagen, M., Wellner, B., Lee, C., Pustejovsky, J.: Machine learning of temporal relations. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, p. 760. Association for Computational Linguistics (2006)
22. Pustejovsky, J.: Personal correspondence (2009)
23. Setzer, A., Gaizauskas, R.: Annotating events and temporal information in newswire texts. In: Proceedings of the Second International Conference On Language Resources And Evaluation (LREC-2000), vol. 31, Athens, Greece (2000)
24. Mani, I., Wellner, B., Verhagen, M., Pustejovsky, J.: Three approaches to learning TLINKS in Time ML. Technical Report CS-07-268, Brandeis University, Waltham, MA, USA (2007)
25. Pustejovsky, J., Stubbs, A.: Natural language annotation for machine learning. O'Reilly Media, Inc. (2012)
26. Verhagen, M.: Temporal closure in an annotation environment. Lang. Res. Eval. **39**(2), 211–241 (2005)
27. Chklovski, T., Pantel, P.: Verbocean: Mining the web for fine-grained semantic verb relations. Proc. EMNLP **4**, 33–40 (2004)
28. Chklovski, T., Pantel, P.: Path Analysis for Refining Verb Relations. In: Proceedings of KDD Workshop on Link Analysis and Group Detection (LinkKDD-04) (2004)
29. Lin, D., Pantel, P.: Discovery of inference rules for question-answering. Natural Lang. Eng. **7**(04), 343–360 (2002)
30. Chambers, N., Jurafsky, D.: Unsupervised learning of narrative event chains. In: Proceedings the Annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics (2008)

31. Boguraev, B., Pustejovsky, J., Ando, R., Verhagen, M.: Time Bank Evolution as a Community Resource for TimeML Parsing. Lang. Res. Eval. **41**(1), 91–115 (2007)
32. Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., Pustejovsky, J.: SemEval-2007 Task 15: TempEval: temporal relation identification. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval) (2007)
33. 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)
34. Tannier, X., Müller, P.: Evaluating temporal graphs built from texts via transitive reduction. J. Artif. Intell. Res. **40**, 375–413 (2011)
35. Kelley Jr, J., Walker, M.: Critical-path planning and scheduling. AFIPS Joint Computer Conferences, pp. 160–173 (1959)
36. Levenshtein, V.: Binary codes capable of correcting spurious insertions and deletions of ones. Probl. Inf. Trans. **1**(1), 8–17 (1965)
37. UzZaman, N., Allen, J.: Temporal evaluation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 351–356. Association for Computational Linguistics (2011)
38. Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Moszkowicz, J., Pustejovsky, J.: The TempEval challenge: identifying temporal relations in text. Lang. Res. Eval. **43**(2), 161–179 (2009)
39. 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)
40. UzZaman, N., Llorens, H., Derczynski, L., Verhagen, M., Allen, J., Pustejovsky, J.: SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In: Proceedings of the SemEval workshop. Association for Computational Linguistics (2013)
41. Minard, A.L., Agirre, E., Aldabe, I., van Erp, M., Magnini, B., Rigau, G., Speranza, M., Urizar, R.: SemEval-2015 Task 4: TimeLine: Cross-Document Event Ordering. In: Proceedings of the workshop on Semantic Evaluation. Association for Computational Linguistics (2015)
42. Llorens, H., Chambers, N., UzZaman, N., Mostafazadeh, N., Allen, J., Pustejovsky, J.: SemEval-2015 Task 5: QA TempEval. In: Proceedings of the workshop on Semantic Evaluation. Association for Computational Linguistics (2015)
43. Bethard, S., Derczynski, L., Pustejovsky, J., Verhagen, M.: SemEval-2015 Task 6: Clinical TempEval. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). Association for Computational Linguistics. Association for Computational Linguistics (2015)
44. Popescu, O., Strapparava, C.: SemEval-2015 Task 7: Diachronic Text Evaluation. In: Proceedings of the workshop on Semantic Evaluation. Association for Computational Linguistics (2015)
45. Boguraev, B., Ando, R.: TimeML-compliant text analysis for temporal reasoning. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI) (2005)
46. Zhang, T., Damerau, F., Johnson, D.: Text chunking based on a generalization of winnow. J. Mach. Learn. Res. **2**, 615–637 (2002)
47. 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 '07, pp. 438–441. Association for Computational Linguistics (2007)
48. Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., et al.: The TimeBank Corpus. In: Proceedings of the Corpus Linguistics conference, pp. 647–656 (2003)
49. ARDA: Aquaint timeml corpus (2006). http://www.timeml.org/site/timebank/timebank.html
50. Mirroshandel, S., Ghassem-Sani, G., Khayyamian, M.: Using syntactic-based kernels for classifying temporal relations. J. Comput. Sci. Technol. **26**(1), 68–80 (2010)

51. Chambers, N., Jurafsky, D.: Jointly combining implicit constraints improves temporal ordering. In: Proceedings of EMNLP, pp. 698–706. ACL (2008)
52. Vasilakopoulos, A., Black, W.: Temporally ordering event instances in natural language texts. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005) (2005)
53. Mani, I., Schiffman, B., Zhang, J.: Inferring temporal ordering of events in news. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pp. 55–57. Association for Computational Linguistics (2003)
54. Reichenbach, H.: The tenses of verbs. Elements of Symbolic Logic. Dover Publications, New York (1947)
55. Kolya, A., Ekbal, A., Bandyopadhyay, S.: Event-time relation identification using machine learning and rules. In: Text, Speech and Dialogue, pp. 117–124. Springer, Heidelberg (2011)
56. Minsky, M.: Logical versus analogical or symbolic versus connectionist or neat versus scruffy. AI Magazine **12**(2), 34 (1991)
57. Kolya, A., Ekbal, A., Bandyopadhyay, S.: Event-event relation identification: a CRF based approach. In: 2010 International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE), pp. 1–8. IEEE (2010)
58. Kolya, A., Ekbal, A., Bandyopadhyay, S.: JU_CSE_TEMP: A first step towards evaluating events, time expressions and temporal relations. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 345–350 (2010)
59. Min, C., Srikanth, M., Fowler, A.: LCC-TE: A hybrid approach to temporal relation identification in news text. In: Proceedings of SemEval-2007, pp. 219–222. ACL (2007)
60. Hagège, C., Tannier, X.: XRCE-T: XIP temporal module for TempEval campaign. In: Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007), pp. 492–495 (2007)
61. Yoshikawa, K., Riedel, S., Asahara, M., Matsumoto, Y.: Jointly identifying temporal relations with Markov logic. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 405–413. Association for Computational Linguistics (2009)
62. Bethard, S., Martin, J., Klingenstein, S.: Finding temporal structure in text: machine learning of syntactic temporal relations. Int. J. Semant. Comput. **1**(4), 441 (2007)
63. Cheng, Y., Asahara, M., Matsumoto, Y.: NAIST.Japan: temporal relation identification using dependency parsed tree. In: Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval '07, pp. 245–248. Association for Computational Linguistics (2007)
64. 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)
65. Marsic, G.: Temporal processing of news: Annotation of temporal expressions, verbal events and temporal relations. Ph.D. thesis, University of Wolverhampton (2011)
66. Puşcaşu, G.: WVALI: Temporal relation identification by syntactico-semantic analysis. In: Proceedings of the 4th International Workshop on SemEval, vol. 2007, pp. 484–487 (2007)
67. Ha, E., Baikadi, A., Licata, C., Lester, J.: NCSU: Modeling temporal relations with markov logic and lexical ontology. In: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval '10, pp. 341–344. Association for Computational Linguistics (2010)
68. Llorens, H., Saquete, E., Navarro, B.: TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In: Proceedings of the SemEval-2010, pp. 284–291. ACL (2010)
69. Llorens, H., Saquete, E., Navarro-Colorado, B.: Automatic system for identifying and categorizing temporal relations in natural language. Int. J. Intell. Syst. **27**, 680–708 (2012)
70. Derczynski, L., Gaizauskas, R.: Using signals to improve automatic classification of temporal relations. In: Proceedings of the ESSLLI StuS (2010)
71. Derczynski, L., Gaizauskas, R.: USFD2: Annotating temporal expressions and TLINKs for TempEval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 337–340. Association for Computational Linguistics (2010)

72. Bethard, S., Savova, G., Chen, W.T., Derczynski, L., Pustejovsky, J., Verhagen, M.: Semeval-2016 task 12: Clinical tempeval. In: Proceedings of SemEval pp. 1052–1062 (2016)
73. Miller, T.A., Bethard, S., Dligach, D., Pradhan, S., Lin, C., Savova, G.K.: Discovering narrative containers in clinical text. ACL **2013**, 18 (2013)
74. Pustejovsky, J., Stubbs, A.: Increasing informativeness in temporal annotation. In: Proceedings of the 5th Linguistic Annotation Workshop, pp. 152–160. Association for Computational Linguistics (2011)
75. Tatu, M., Srikanth, M.: Experiments with reasoning for temporal relations between events. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, pp. 857–864. Association for Computational Linguistics (2008)
76. Dang, H., Lin, J., Kelly, D.: Overview of the trec 2006 question answering track. In: Proceedings of the Text Retrieval and Evaluation Conference (2008)
77. Howald, B., Katz, E.: On the explicit and implicit spatiotemporal architecture of narratives of personal experience. Spatial Information Theory, pp. 434–454 (2011)
78. Ji, H., Grishman, R., Dang, H., Li, X., Griffit, K., Ellis, J.: Overview of the TAC2011 Knowledge Base Population Track. In: Proceedings of the Text Analytics Conference (2011)
79. Burman, A., Jayapal, A., Kannan, S., Kavilikatta, M., Alhelbawy, A., Derczynski, L., Gaizauskas, R.: USFD at KBP 2011: Entity linking, slot filling and temporal bounding. In: Proceedings of the Text Analytics Conference (2011)
80. Chambers, N., Wang, S., Jurafsky, D.: Classifying temporal relations between events. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp. 173–176. Association for Computational Linguistics (2007)
81. Mirroshandel, S., Ghassem-Sani, G.: Temporal relation extraction using expectation maximization. In: Proceedings of RANLP (2011)
82. Mirroshandel, S., Ghassem-Sani, G., Nasr, A.: Active learning strategies for support vector machines, application to temporal relation classification. In: Proceedings of 5th International Joint Conference on Natural Language Processing, pp. 56–64 (2011)
83. Mirroshandel, S., Ghassem-Sani, G.: Temporal relations learning with a bootstrapped cross-document classifier. In: Proceeding of the 19th European Conference on Artificial Intelligence, pp. 829–834 (2010)
84. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: the 90% solution. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, pp. 57–60. Association for Computational Linguistics (2006)
85. Lapata, M., Lascarides, A.: Learning sentence-internal temporal relations. J. Artif. Intell. Res. **27**(1), 85–117 (2006)
86. Gaizauskas, R., Harkema, H., Hepple, M., Setzer, A.: Task-oriented extraction of temporal information: the case of clinical narratives. In: Thirteenth International Symposium on Temporal Representation and Reasoning, TIME 2006, pp. 188–195. IEEE (2006)
87. Bramsen, P., Deshpande, P., Lee, Y., Barzilay, R.: Finding temporal order in discharge summaries. In: AMIA Annual Symposium Proceedings, vol. 2006, p. 81. American Medical Informatics Association (2006)
88. Bethard, S., Martin, J.: CU-TMP: temporal relation classification using syntactic and semantic features. In: Proceedings of the 4th International Workshop on Semantic Evaluations. SemEval '07, pp. 129–132. Association for Computational Linguistics, Stroudsburg, PA, USA (2007)
89. Costa, F., Branco, A.: Temporal relation classification based on temporal reasoning. In: Proceedings of the International Conference on Computational Semantics. Association for Computational Linguistics (2013)
90. Hovy, D., Fan, J., Gliozzo, A., Patwardhan, S., Welty, C.: When did that happen?: linking events and relations to timestamps. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 185–193. Association for Computational Linguistics (2012)
91. Laokulrat, N., Miwa, M., Tsuruoka, Y.: Exploiting timegraphs in temporal relation classification. In: TextGraphs-9 (2014)

92. Puşcaşu, G.: Discovering temporal relations with TICTAC. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2007) (2007)
93. Lee, C., Katz, G.: Error analysis of the TempEval temporal relation identification task. In: SEW-2009 Semantic Evaluations: Recent Achievements and Future Directions, pp. 138–145 (2009)
94. Velupillai, S., Mowery, D.L., Abdelrahman, S., Christensen, L., Chapman, W.W.: BluLab: Temporal information extraction for the2015 clinical TempEval challenge. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). Association for Computational Linguistics. Association for Computational Linguistics (2015)
95. Scheuermann, A., Motta, E., Mulholland, P., Gangemi, A., Presutti, V.: An empirical perspective on representing time. In: Proceedings of the seventh international conference on Knowledge capture, pp. 89–96. ACM (2013)
96. Derczynski, L.: Representation and Learning of Temporal Relations. Proceedings of the 26th International Conference on Computational Linguistics. Association for Comptuational Linguistics (2016)