Determining the Types of Temporal Relations in Discourse

The University Of Sheffield.

Leon R.A. Derczyński

University of Sheffield

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

May 2013

This thesis is dedicated to
Camilla,
my supervisor Rob Gaizauskas,
my colleagues,
my examiners Mark Steedman and Yorick Wilks,
and everybody else who has been ignored far too much
(or too little)
during its creation;

especially Ahmet Aker, Nikos Aletras, Emma Barker, Kalina Bontcheva, Danica Damljanovic, David Elson, David Field, Adam Funk, Douwe Gelling, Mark Greenwood, David Guthrie, Mathew Hall, Mark Hepple, Héctor Llorens, Diana Maynard, Philippe Muller, Daniel Preotiuc, James Pustejovsky, Dominic Rout, Estela Saquete, Jannik Strötgen, Kumutha Swampillai, Naushad UzZaman and Marc Verhagen.

Acknowledgements

The author would like to acknowledge the UK Engineering and Physical Science Research Council's support in the form of a doctoral studentship.

Abstract

The ability to describe the order of events is crucial for effective communication. It is used to describe causality, to plan and to relay stories.

Event ordering can be thought of in terms of binary temporal relations which hold between event pairs or pairs of times and events. Complex event structures can be modelled as compositions of these pairs.

Temporal relations can be expressed linguistically in a variety of ways. For example, one may use tense to describe the relation between the time of speaking and other events, or use a temporal conjunction to temporal situate an event relative to time.

In the area of automatic temporal information extraction, determining the type of temporal relation between a given pair of times or events is currently the hardest task. Very sophisticated approaches have yielded only small improvements over initial attempts.

Rather than develop a generic approach to event ordering through relation typing, a failure analysis informs grouping and segmentation of these temporal relations according to the type of information that can be used to temporally relate them. Two major sources of information are identified that provide typing information for two segments: relations explicitly described by a signal word, and relations involving a shift of tense and aspect.

Following this, we investigate automatic temporal relation typing in both these segments, presenting results, introducing new methods and a generating set of new language resources.

Contents

List of Tables

List of Figures

List of Abbreviations

ATC	AQUAINT/TimeBank Corpus
CRF	Conditional Random Fields
CSV	Comma-Separated Values
DCT	Document Creation Time
IAA	Inter-Annotator Agreement
IE	Information Extraction
ILP	Integer Linear Programming
NER	Named Entity Recognition
NLP	Natural Language Processing
NLTK	Natural Language ToolKit
PoS	Part of Speech
QA	Question Answering
RRM	Robust Risk Minimisation
RTMML	Reichenbach Tense Model Markup Language
SVM	Support Vector Machine
TimeML	Time Markup Language
TIMEX	Temporal Expression
TLINK	Temporal Link
XML	eXtensible Markup Language
XV	Cross-Validation

Chapter 1

Introduction

> Le temps mûrit toute choses; par le temps toutes choses viennent en évidence; le temps est père de la vérité.
> *Time ripens all things; with Time all things are revealed; Time is the father of truth.*
>
> *Gargantua and Pantagruel*
> Francois Rabelais

1.1 Setting the Scene

Humans developed natural language to communicate; over past millennia, it has been the most efficient form of transferring the majority of information between individuals. With the advent of computing, large amounts of natural language text are stored in digital format. Computational linguistics helps link the significant power of the computer with the efficiency of communicating in natural language. This thesis is situated within the field of computational linguistics, which concentrates on automatic processing of human language.

Within computational linguistics, this thesis fits in the sub-field of information extraction. This sub-field concentrates on the automatic identification of specific information about entities, relations or events from natural language discourse (Gaizauskas and Wilks, 1998). It has developed to a point where information such as person-relation-data triples (such as *Helle Thorning-Schmidt : job : Prime Minister*) can often be reliably identified (Carlson et al., 2010). The thesis concentrates on extracting information about temporal relations, which is a challenging and difficult problem.

There is a strong need to understand time in discourse. Time is critical to our ability to communicate plans, stories and change. Further, much of the information and many of the assertions made in a text are bounded in time. For example, the sky was not always blue; George W. Bush's presidency was confined to an eight-year interval. An understanding of time in natural language text is critical to effective communication and must be accounted for in automatic processing and understanding of discourse.

Being able to identify times and events, the basic entities of temporal reasoning, within natural language discourse is not enough to understand its temporal structure. Events cannot indepen-

dently be placed onto a calendar scale. To situate events temporally, they must be related to other events or to times. These relations, however interpreted, are the information describing the temporal structure of a text (corresponding to the C-series of McTaggart (1908): the fixed ordering of events); they allow one to situate an event in terms of times or other events and to describe the complexity of some event structures (for example, events with sub-events (Pustejovsky, 1991)).

Determining temporal relations is critical to understanding the temporal situation of events described in discourse. Automatic extraction of temporal relations has proven difficult, though it is something that human readers can perform very readily. Human readers are likely to have access to information from a given discourse as they read it and from experience in the form of world knowledge. That we can identify the nature of temporal relations easily suggests that the information required for temporal relation extraction is contained either in discourse or in world knowledge.

The task can be broken into two parts, given any document: identifying which phrases correspond to temporal entities in the document, such as times and events; and identifying how these entities are related to each other. Example 1.1 contains a selection of words and phrases of temporal relevance.

(1.1) **Nov. 17, 2006**$_1$ China's first ever space textbook **declassified**$_2$ and published.

> Qian Xuesen's manuscript entitled "A General Introduction to the Missile" hit the shelves in Beijing on **Friday**$_3$, 50 years **after**$_4$ Qian first used it to **teach**$_5$ 156 university students, China's first generation of space scientists.

There are two times, (1) and (3); the first (1) is acting as the document's creation time. There are also examples of events, (2) and (5). Finally, there is a temporal signal (4), which explicitly describes the *type* of temporal relation that holds between a time (3) and an event (5).

Separate bodies of work focus on extracting *events* from texts and also on identifying and interpreting textual references to *times*. These have reached high recognition accuracies in both event and time recognition (see Sections 2.2.3 and 2.3.3). However, identifying the types of relations that hold between these entities is still a difficult problem. After many years' effort, performance levels currently reach about 70% accuracy (see Section 3.5).

Rather than attempt to further existing work on annotating events or times, the topic of this thesis is the temporal relations that hold between them. The task of determining which event and time pairs ought to be linked is referred to as the **relation identification** task. The problem of automatically determining the type of temporal relations that exist between a given pair of events or times is knowns as the **temporal relation typing** task. This thesis focuses on the temporal relation typing task.

Temporal relation typing consists of describing the kind of temporal ordering or relation between a pair of temporal entities, which are in themselves events or times. That is, deciding which order events happen in, according to the text. In Example 1.1, there are two events: entity 2, *declassified*, and entity 5, *teach*. To type the temporal relation between these two, we have to decide in which order they occur, according to the text. For this case, we would say that the *declassified* event happens <u>after</u> the *teach* event.

Finding that relation type just from the text is the relation typing task, and the focus of this work.

Once we can automatically find and label the temporal relations between events and times in a discourse, powerful techniques become available for improving human-computer interaction and automatic processing of information stored in natural language. For example, systems can perform better in answering questions put to them in natural language; it becomes easier to create better summaries of discourse; work on forensic applications – such as building timelines of witness statements in the event of a crime or transportation disaster – can be automated; one may construct stories using potentially incomplete accounts from multiple sources; and temporal information extraction could be used in everyday communication, to organise events and create calendar appointments automatically from personal communication. Work to solve the overall temporal information extraction problem continues in many specific areas of temporal information extraction, such as time and event recognition.

Considering temporal information allows better-informed processing of natural language. Understanding of temporal relations helps build rich and accurate models of information from discourse. For example, Wang et al. (2010) use manually-added temporal information to augment kernel-based discourse relation recognition. Further, temporal expressions and relations often help not only with segmenting texts but also with relating matching segments (Bestgen and Vonk, 1999; Bramsen et al., 2006; Jean-Louis et al., 2010). Good temporal information extraction also provides clear benefits to summarisation and automatic biography systems (e.g. Filatova and Hatzivassiloglou (2004)), which require temporal information in order to determine clusters of events and a correct story order. Models of time in language have also been of use in machine translation (Horie et al., 2012). Without an understanding of temporal relations in text, question answering systems cannot tackle any "when" questions or distinguish history and conjecture from current world state. In fact, every assertion is bounded in time and these bounds must be recognised in order to reason about events and world knowledge over time.

An example of the applications of temporal information in the NLP task of question answering follows. A system may be asked *"When was the current president of the USA elected?"*. Typically in question answering scenarios, a set of texts is provided as a basis for determining the correct answer. In this case, such texts may mention many presidents and have creation dates spread over many years. While statistical approaches using document timestamps have some success in answering temporal queries, better language understanding can provide reliable and precise results once the temporal information in discourse can be accurately interpreted. In this case, there may be multiple challenges: the current time must be determined, the current president must be identified, and the election event related to the identified president. Alternatively, one may simply look for the most recent match for this election event. Either way, an understanding of time in text is required. To answer this correctly, systems may require not only the ability to situate a question in time and relate events described in a discourse to its creation date, but also distinguish mentions of historical events from more recent ones. Such a level of sophistication is not yet present in the state of the art of automatic temporal relation extraction.

Finally, prototype industrial applications have already been created that rely upon accurate

relation of events and times in text. Carsim (Johansson et al., 2005) extracts relations from car accident reports and uses the resulting information to construct 3D visualisations of potential stories leading to an accident. In the legal industry, temporal relation extraction is a critical part of verifying and merging independent accounts related to investigations (Howald and Katz, 2011). Lastly, medical record processing involves a temporal aspect, allowing systems to automatically order events in a case history (Savova et al., 2009; Jung et al., 2011).

The remainder of this chapter has three functions. The scope of the task attempted by this thesis is stated first, followed by a description of the scientific contributions that it makes. The chapter concludes with a brief outline of the thesis structure.

1.2 Aims and Objectives

How can we identify the information needed to determine the nature of a temporal relation and then use it to help automatically determine the nature of temporal relations? The previous section has described the kinds of temporal primitive used to convey temporal structure through natural language text, with a two-part general structure of (a) times and events and (b) relations between them. The focus of this thesis is the latter – relations. The automated processing of relations can be decomposed into two tasks. Firstly, the relation endpoints (individual textual references to times or events) are identified. Secondly, the nature of this binary relation is determined and described using a type from a pre-defined set of relations (containing concepts such as precedence, inclusion and identity).

For this thesis, we focus on the second task – determining the types of temporal relations – and do not attempt to determine which endpoints should be related (relation identification). Discovering which events or times might be related to each other is a difficult task to define; every time and event has temporal bounds, and so a temporal relation of some kind exists between all of them, though this may not be critical to the story told by any given discourse. Further, human-annotated datasets are available where the subjectively most salient binary temporal relations have already been identified. This allows us to focus on the pertinent and difficult task of determining the nature of relations, without worrying about the ill-defined task of how we should choose which binary relations to investigate.

This thesis reports on an evidence-backed, rational investigation into the difficulties of automatically understanding the temporal structure of a discourse. The key aims are:

1. To identify new information sources useful for temporal ordering;
2. To provide improved methods for extracting temporal ordering information;
3. To suggest avenues of further research in the area.

The work is constrained to English-language text in the newswire genre.

1.3 Contribution

This thesis details an investigation into automatically determining temporal relations in text. This comprises a data-driven analysis of the temporal relation characterisation problem followed

by two approaches to improving automatic performance at relation typing. The significant and novel contributions are described below.

The analysis draws upon new findings based on openly available datasets. It comprises the first analysis of temporal relation typing results from the Tempeval-2 evaluation task (Verhagen et al., 2010), where many teams tested state-of-the-art temporal information extraction systems on a common set of data, including an attempt to automatically determine the nature of temporal relations. The analysis results in the definition of a "difficult" temporal relation, and identifies a consistently difficult set of temporal relations. These are the relations that must be conquered for research in the area to progress. Based on this set of difficult relations, there are quantitative and qualitative failure analyses. These explore a variety of linguistic phenomena related to temporality, and their prevalence among difficult temporal relations. Emerging from this new and detailed analysis, the section concludes with the presentation of evidence-based suggestions of directions for further research. Two of these are investigated in the following chapters.

The first approach for improving relation extraction is based on words and phrases that explicitly state the nature of a temporal relation – temporal signals. It begins with an empirical confirmation of signals as supporting temporal relations. Such confirmation is followed with the introduction and demonstration of a new technique, using signals to help in temporal relation typing, that achieves over 53% error reduction when compared to the state of the art. As these signals are of great utility, a corpus-driven characterisation of temporal signals is given, the first of its kind, including statistical results and a formal definition of this closed class. It is found that these signals exhibit two kinds of polysemy, and quantitative results are presented describing both of these. Firstly, a linguistic polysemy where the signal words and phrases have both temporal and non-temporal meanings. Secondly, a temporal polysemy, where the same signals may have differing temporal interpretations depending on their context. After this characterisation, existing resources are curated and augmented to create a temporally annotated corpus with high quality signal annotations, which is presented as a new linguistic resource. Having been shown to be useful but ambiguous, we introduce a two-stage process for annotating temporal signals. The initial part of this is an effective method for automatic identification of signals, which is similar to a specialised word sense disambiguation task. The remaining part of the process is an effective method for, given a word or phrase that acts as a temporal signal, associating signals with their event or time arguments, thus connecting a signal to a binary temporal relation. This final part entails the definition of a new discourse-based task, and proposes multiple approaches, arriving at a successful initial solution. The approach concludes with a demonstration that these new techniques for identification of signals and association with their arguments, coupled with our approach for using them to support relation extraction, improves on the overall characterisation of temporal relations within a given discourse.

The second approach for improving relation extraction centres upon a framework of tense and aspect than can deterministically provide basic temporal ordering information between some events and times. Because the framework is not directly applicable to existing resources, two interpretations for the framework are put forward. The subsequent investigation begins with the first corpus-driven validation of this established framework of tense and aspect. The validation

empirically demonstrates that the model is consistent with a gold standard temporally annotated corpus, but that finding which events and times are connected is an open problem. To confirm the results of this validation, the model's predictions are for the first time integrated into machine learning approaches for describing temporal relations. This gives improvements for event-event relation typing. It is also shown that the model has utility for determining the nature of relations between times and events. A technique is presented for using the model to automatically classify the temporal relations between times and events. The problem of determining which events and times to consider for connection under this framework is shown to be a limiting factor in its application. Finally, an ISO-compliant mark-up for integrating this model with established temporal annotation schemas is presented to aid further work using the model, parts of which have been shown to be helpful for event-time relation extraction and to be critical to the understanding of some other temporal phenomena.

1.4 Structure of the thesis

This thesis details a plan for improving temporal relation typing and describes its outcome. It finds that to understand the temporal ordering of events described in text we cannot rely on a single set of information for every relation. Instead, we need to draw upon multiple heterogeneous information sources. A set of these information sources is identified and techniques introduced for exploiting them to improve temporal relation typing. Two of these information sources – one related to explicit temporal signal words and phrases, another related to tense and aspect – are explored in depth.

The remainder of this document is divided into three major sections. Early chapters introduce the field and prior work. The next set of chapters comprise the core of the work and describe the experimental approach and its results. Finally, an overview is given, discussing applications of temporal relation typing and providing conclusions. Appendices contain supplementary material.

Chapter 2 describes necessary theoretical and computational background. Chapter 3 explores related work on the specific task of temporal relation extraction, concluding with the current state of the art. It also briefly introduces current event and time extraction systems.

The temporal relation extraction problem is detailed in depth in Chapter 4, which also includes failure analysis of a set of temporal relation classifiers and outlines our approach for the remainder of the experimental work. Chapter 5 introduces temporal signals, showing how they are useful for relation typing and how they can be automatically annotated in a helpful manner. Chapter 6 details a model of tense which is used to improve relation typing between verb events.

Finally, Chapter 7 provides a formal summary of the thesis and a discussion of promising future directions for temporal relation typing and the overall task of temporal information extraction.

1.5 Previously published material

Some parts of this thesis overlap with published work. The extent to which material in this thesis occurs in published literature is as follows.

This chapter and Chapters 2, 3 and 4 are previously unpublished.

In Chapter 5, the background material (Sections 5.2, 5.4) has been presented in Derczynski and Gaizauskas (2011a) and one initial experiment (Section 5.3) in Derczynski and Gaizauskas (2010c). In Chapter 6, the notion of temporal context (Section 6.3.4) and the proposed annotation scheme (Section 6.6) are both introduced in a prototypical form in Derczynski and Gaizauskas (2011b), and the advanced schema validation (Section 6.4.2) is the topic of Derczynski and Gaizauskas (2013a).

The remainder of the thesis is all previously unpublished.

Chapter 2

Events and Times

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

2.1 Introduction

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

If we are to have a computer reason about times and events, we need to know about time in language. Time in language can be broken down into three primitives: times, events and temporal relations (Moens and Steedman, 1988). Viewing the temporal structure of a discourse as a graph, the times and events are the nodes and the relations the arcs. In this chapter, we introduce the nodes – events and times.

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

Finally, given this understanding, it is possible to build systems for some automatic temporal processing. There are approaches to detecting times and events, to determining event durations (Pan et al., 2006; Gusev et al., 2011) and to typing the relation between two events (Mani

and Schiffman, 2005; Mani et al., 2007; Bethard et al., 2007b).

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

2.2 Events

The Oxford English Dictionary defines an event as *"a thing that happens or takes place, especially one of importance"*. This definition could be broken down into occasions, actions, occurrences and states. However, the occasions, actions, occurrences and states are used in natural language more widely than this definition permits; there are often mentions of negated events, conditional events or modal events, which cannot be said to certainly "happen or take place" (Pustejovsky, 1991). Further, events can be composed of many sub-events: for example, the Arab Spring lasted months and included multiple revolutions, each of which had a long history, a complex set of story threads all happening in parallel, a culmination and an aftermath. Events may be represented by a variety of lengths of expressions, ranging from document collections (Ritter et al., 2012) to single tokens. For the purpose of this thesis, the description of events from TimeML (a temporal markup language, Pustejovsky et al. (2004)) is adopted, as follows:

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

Given that they may describe an action or transition, events are often expressed by verbs (*"The bus stopped suddenly"*). A nominalised event is an event that is represented by a noun phrase. For example, one might mention *the explosion*, which is a noun that describes an event. Events may also be expressed by statives, as in *the man was an idiot*); by predicatives, as in *Elizabeth is queen*; by adjectives, in *the storm is active*; and by prepositional phrases, such as in *soldiers will be present in uniform.* A further discussion of events and states can be found in Steedman (1982).

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

2.2.1 Types of Event

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

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

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

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

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

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

Aspectual Finally, aspectual events are those expressions which describe certain parts of the life of an event, such as its beginning, culmination, continuation and so on. For example, *The scientists were starting to show signs of exhaustion.* See also Vendler (1957).

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

2.2.2 Schema for Event Annotation

Given definitions of events and a need to process them automatically, some kind of formal method of describing events must be introduced. For this, and for temporal annotation over the remainder of this thesis, we adopt TimeML. TimeML (Pustejovsky et al., 2004) is an XML-style markup for temporal information in natural language texts and has become an ISO standard. An overview of the syntax and annotation guidelines can be found online.[1]

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

It is important to determine exactly what to annotate. Events may have actors, for example, and may be expressed using auxiliaries, prepositional phrases, negation and modal signifiers, and so on. The contiguous sequence of words that describes an event is called the event chunk. The single most important word within this chunk – the one that critically defines the event, such as the dominant verb – is the chunk (or event) head. In TimeML, `<EVENT>` annotations are applied to the shortest possible phrase that could describe the event; e.g., its head. See Example 2.1 from the TimeML 1.2.1 annotation guidelines.

(2.1) *He would not have been going to permit anything like that.*

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

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

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

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

2.2.3 Automatic Event Annotation

Task Description

Complete event annotation comprises **event recognition** (determining which expressions denote events) and **event classification** (characterising events once found). Recognition concerns determining which words or phrases can be marked up as being events. Event classification involves determining the "class" of a particular event (such as an action or a state) according to a schema such as that presented in Section 2.2.1. Performing both tasks together is generally harder than just recognising where events lie in text (Boguraev and Ando, 2005).

Evaluation

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

Identifying event extents Event recognition is the task of identifying and delimiting event phrases. A perfect system will mark all events, determining their textual bounds correctly and not mark any text that is not an event. Evaluation metrics should thus reward systems for both finding events and also for not finding non-events. Precision and recall fit these requirements and are often used to evaluate event recognition (Verhagen et al., 2010). A brief description of precision and recall follows.

Recall is the proportion of existing items that have been identified by a system; a system that returns one event in a document that actually contains ten has a recall of 10%. However, a system that marks everything as an event is bound to find all events and has a recall of 100%. To balance this, one may introduce precision. Precision is the proportion of returned items that are correct;

returning just one correct item and no others gives 100% precision, but returning everything where there are only a few events will generate a low precision score.

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

$$recall\ R = \frac{W \cap E}{W} \tag{2.2}$$

$$precision\ P = \frac{W \cap E}{E} \tag{2.3}$$

Relations between precision and recall are discussed by Buckland and Gey (1994). It is common to combine the two with a harmonic mean such as F-measure (van Rijsbergen, 1979). The formula is as follows:

$$F_\beta = (1 + \beta^2)\frac{PR}{\beta^2 P + R} \tag{2.4}$$

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

Approaches

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

Boguraev and Ando (2005) cast TimeML EVENT recognition as a machine learning chunking problem. Text is treated as a sequence of tokens to which labels are assigned which describe chunk boundary information; three labels are possible – E for an end of a chunk, I for a token inside a chunk and O for "any token outside a target chunk". Features are then generated based on capitalisation, n-gram, part of speech, chunk type and head word information, similar to a word-profiling approach to entity recognition (Ando, 2004). Following this, recognising EVENT extents in the Wall Street Journal is 77-80% accurate (F-measure). This figure drops to 61-64% accuracy for the joint task of recognising event extents and then correctly assigning TimeML classes to these events. The difference shows that the event classification task is non-trivial, having similar success rates to the approach used here for event recognition (e.g. around 75-80%).

EVITA (Saurí et al., 2005), included in the TARSQI toolkit (Section B.3.1), employs different strategies for dealing with verb, noun and adjective events. It uses both machine learning and knowledge-based techniques. Verbs are filtered based on the verbal chunk head, modal auxiliaries and event polarity. Nouns are filtered against a look-up table and sense disambiguation lookup (to repeat the example from the paper, a noun in WordNet synset *phenomenon* is not an event if is it also subsumed by the synset *cloud*). Finally, adjectives are only tagged as events if they have already been used as such by a gold standard source (such as TimeBank). EVITA reaches 80%

F-measure when recognising verbal events in TimeBank 1.2, which is comparable to IAA scores from that corpus' creation.

The most recent efforts in automatic TimeML event annotation focus on machine learning approaches incorporating information about semantic roles, reaching F-measures of over 0.80. The leading tool, TIPSem-B (Llorens, 2011), incorporates semantic role information into its CRF-based event annotation approach. It is openly available for download.[2]

2.3 Temporal Expressions

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

Linguistic characterisation of temporal expressions has led to discussion and observations regarding their usage and situation. Hitzeman (1997) find that time expressions are often used as discourse segmentation markers and highlights their potential ambiguity. They find that the interpretation of a given temporal expression depends on its syntactic position. Similarly, Bestgen and Vonk (1999) show that temporal expressions used as adverbials help set the scene for a sub-part of discourse, providing a context and a timeframe and are helpful discourse segmentation markers, improving discourse comprehension. Cohen and Schwer (2012) perform multi-lingual characterisation of temporal markers, describing such expressions as comprising three parts: the size of the temporal segment, the distance from a temporal centre (e.g. a reference point, Section 6.3) and an orientation such as future or past.

For this thesis, a "temporal expression", or **timex**, is any expression that denotes a moment, interval or other temporal region without having to rely upon an event. Each interval is composed of two points between which it obtains. For example, *24th August 1997*, *two weeks* and *now* are all temporal expressions; *after the storm* is not. Hobbs and Pan (2004) define a "proper interval" as one where the start point is before the end point. Under this definition, this thesis considers only "proper interval" as intervals; that is, no minimum atomic duration is recognised, and there is no quantisation of time into chronons. Rather, temporal entities are described by infinitesimal points that bound them.

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

2.3.1 Temporal expression types

Before describing algorithms that can identify and anchor time expressions, we will briefly equip the reader with a short summary of types of time expression. Most papers that cover this topic, using varying nomenclatures, settle on a small set of different types of time expressions defined

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

by their authors (Mani and Wilson, 2000; Ferro et al., 2005; Ahn et al., 2005; Han et al., 2006; Derczynski et al., 2012). These types can generally be mapped onto one of the following distinct classes.

- **Absolute** — Where the text explicitly states an unambiguous time. Depending on the granularity of the interval, the text includes enough information to narrow a point or interval directly down to one single occurrence. This is in contrast to a time which, while precise and maybe easy for humans to pin onto a calendar, relies on an external reference. For example, the interval *Thursday October 1st, 2009* would be considered absolute, but *The week after next* would not – the information is not all explicit or held in the same place; this latter expression implies reliance on some external reference time. Absolute expressions are sometimes also known as fully-qualified time expressions.
- **Deictic** — Cases where, given a known time of utterance, one can determine the period being referred to. These time expressions, specify a temporal distance and direction from the utterance time. One might see a magazine bulletin begin *Two weeks ago, we were still in Saigon.*; this expression leaves an unclear implicit speech time, which one could safely assume was the date the article was written. More common examples include *tomorrow* and *yesterday*, which are both offset from speech time; to describe this using Reichenbach's model (Section 6.3), deictic temporal expressions represent situations where reference time and speech time are the same.
- **Anaphoric** — Where speech and reference time are not at the same point. Anaphoric temporal expressions have three parts – temporal distance (e.g. 4 days), temporal direction (past or future) and an anchor that the distance and direction are applied from. The anchor, for anaphoric temporal expressions, is the current reference time as per Reichenbach's model (Section 6.3). Example phrases include *the next week*, *that evening* or *a few hours later*, none of which can be anchored even when their speech time is known.
- **Duration** — A duration describes an interval bounded by a start and an end, where the distance between the two is known, but the expression itself is not placeable on any external time system (like a calendar). Durations generally include a time unit as their head token; for example, *ninety minutes* is a single duration timex. This type of temporal expression is easily confused with deictic expressions; to use Ahn's example (Ahn et al., 2005),

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

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

2.3.2 Schema for timex annotation

Temporal expressions are often inherently vague, and typically only communicated only to the level of precision that the speaker requires in order to convey their point coherently. As a result,

it is difficult to develop a precise, discrete knowledge representation form for timexes – the classic AI problem of building machine-readable forms from qualitative concepts. Bearing this in mind, approaches to timex annotation have been developed.

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

To this end, any timex annotation schema has to account for describing both the extents of the expression and the value of the expression itself. Today, the two prevailing standards for timex annotation are TIMEX2 and TIMEX3. These standards evolved through the MUC (Chinchor and Robinson, 1997) exercises and TERN (Boguraev and Ando, 2005) through TIMEX to the two most recent incarnations. Both are XML-based and cater for the timex classes of duration, time, date and set.

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

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

TimeML

TimeML (Pustejovsky et al., 2004) is an XML-based language for temporal annotation. It allows annotation of events and times, with a rich format for each, as well as thorough provision of links to capture relations between events and times:

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

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

[3] ISO WD 24617-1:2007

corpora, as well as multiple iterations of the language and the production of processing tools that can parse the markup.

TimeML does not employ the Allen interval relations, but instead uses its own set, based on Allen's earlier work (Allen, 1983, 1984). Notably, TimeML has no OVERLAPS relation, or way of expressing it. This is clarified in TimeML-strict Derczynski et al. (2013). A fuller introduction to TimeML can be found in Pustejovsky et al. (2005).

ISO-TimeML (Pustejovsky et al., 2010) is a LAF and TEI compatible iteration of TimeML. It permits stand-off annotation, where the SGML annotations do not clutter text by being inline and has a more elegant method of instantiating events. The formal standard is recognised by the ISO and maintained by an active working committee.

TIMEX3

TIMEX3 stipulates the annotation of smaller strings than TIMEX2 (Ferro et al., 2005) and is intended for use alongside mechanisms for annotating temporal links and events. TIMEX2 permits longer expressions, including event-based timexes which are anchored not to absolute scales but to events described in the text; the rationale behind this is that TIMEX2 was not designed for use in an environment that included event annotations, whereas TIMEX3 is intended to be used as part of the TimeML annotation scheme. Section A.2.6 further details the differences between the two standards and describes an approach for converting data from the TIMEX2 to TIMEX3 standard.

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

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

The `value` field may take the form an ISO8601-format date, a `P` followed by a numeric quantity and unit symbol to denote a period, or one of a number of special anaphoric-based values such as `PRESENT_REF`. Its format is not trivial and the TIDES/TimeML documentation are the best resources for its description (Pustejovsky et al., 2004; Ferro et al., 2005). For the scope of this thesis, we generally consider TIMEX3 in the context of TimeML, as we are interested in an annotation schema that covers not only temporal expressions but also events and temporal relations. Other attributes of TIMEX3 annotations include:

- Function in document, to denote special timexes such as the document creation point;
- Type, to capture the timex class;
- Modifier for adding information that cannot be added to the value, such as qualitative information (e.g. *"the dawn of 2000"* would be marked as the year 2000 with a modifier of `start`);

- Quantifier and frequency for describing the repetition pattern of a set timex; for example, *"every other Sunday"* would have a value of `P1W` (period of 1 week) and a quantifier of `every other` – and *"twice a day"* has value `P1D` with a frequency of 2.

TCNL

TCNL (Han et al., 2006) is "a compact representational language" for working with time expressions. A set of operators and labels are defined, which can be combined to produce various offsets or absolute expressions. For example, TCNL looks elegant for simplistic temporal relations; $\{tue, < \{|25\{day\}|@\{dec\}\}\}$ for *Tuesday before Christmas*, or $\{friday, < now\}$ to represent an earlier Friday. A calendar model, working with different levels of granularity, is used to help anchor times. Weeks and months, for example, have different durations and do not share synchronised boundaries, but both – when combined with an integer – can define a solidly bounded absolute interval; e.g. Week 34 2008, or January 2012.

Its authors suggest that TCNL has benefits over TOP (Androutsopoulos, 1999), TIMEX2 and TIMEX3/TimeML; namely, that TCNL is calendar-agnostic, focuses on intensional meaning of expressions (which are allowed in TimeML, but not compulsory and not used in the two largest TimeML corpora), shows contextual dependency by using references such as focus and that its type system makes granularity conversion transparent.

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

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

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

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

Further examples in TCNL and a reference guide to the language, can be found in (Han, 2009).

2.3.3 Automatic timex annotation

Task description

As with events, extracting timexes can be decomposed into a multi-part task. In this case, the principal parts are determining which words and phrases in a document comprise timexes and then assigning various attribute values to that phrase. Once identified, a temporal expression may be converted to a fully specified date or interval. Existing work has investigated the task of "anchoring" or "normalising" temporal expressions; that is, taking tokens from a document and mapping them to an absolute temporal scale, using an agreed notation. For example, while the single 24-hour period that the expression *next Thursday* refers may be immediately clear to us at any given time, some processing is required on the part of a computer to map this into a complete time (specifying at least a year and day). It is also important to choose the correct granularity for

temporal expressions; *next day* refers loosely to the contents of a 24-hour period, not to a period of precisely 86400 seconds occurring between two local midnights (or however many caesium decay events, in SI terms).

Automatic timex annotation is typically a three-stage process. Firstly, one must determine the extents of a temporal expression. This stage may be evaluated using conventional precision and recall measures. Secondly, the timex should be interpreted (Mazur and Dale, 2011), converting it to a representation according to an established convention. This includes assigning both an expression type and value, which can be evaluated with string matching for strict evaluation. Thirdly and optionally, the timex may be anchored to a time scale, which involves mapping it to a specific time, date, or range of times and dates.

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

Evaluating temporal expression annotation

Precision and recall are suitable for evaluating temporal expression recognition (see Section 2.2.3). Temporal expressions can also be broken down into one of many classes and may be interpreted or even anchored to a calendar. To evaluate temporal expression typing, a simple "proportion correct" or accuracy metric works well. Interpretation and anchoring efforts can be compared verbatim to a gold standard to assess accuracy. One must also choose whether or not to allow equivalent matches to be considered as equal. For example, the TIMEX3 values `P1D` and `P24H` both correspond to a duration of a day and may be considered equivalent. However, if one prefers an annotation that matches the exact language used in a document, it may be argued that *"one day"* should only be given a value of `P1D` and that `P24H` is more representative of text like *"24 hours"*. Any timex evaluation needs to take a stance on these issues.

Timex annotation systems

Rule based systems are frequently employed in approaches to these tasks, because plenty of set phrases are used to describe time and they employ a simple grammar. In fact, one very successful approach to normalising week days is entirely rule-based (Mazur and Dale, 2008). This attribute of temporal expressions means that finite state grammars can be used for Timex recognition (Ando, 2004; Boguraev and Ando, 2005). In the case of the first paper, a rule-based system was completed by interleaving finite state grammars with named entity recognition, in order to enable temporal expressions in linguistic units, as opposed to lexical ones. This enables the identification of events and associations that are semantically present in a sentence but not immediately obvious from its construction.

Some systems, such as GUTime (Mani and Wilson, 2000), rely heavily on a rule-based approach to spotting sequences of tokens, as there are many temporal expressions present in the English language that can be identified and anchored in this way. Named entity recognition (NER) has also been used to identify times in text (Grover et al., 2010).

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

Rule-based, machine learning-based and hybrid systems all performed well at timex recognition. For English the timex extent recognition performance F-measure ranged from 0.26 to 0.86, with an average of 0.78. The best performance was with F1 of 0.86; seven systems reached F-measures of 0.84-0.86. This is promising, though by no means a solution to the timex recognition problem. Timex classification was performed best by a TIMEX2 transduction system with accuracy 0.98 (Saquete, 2010), though all but two systems attempting timex classification reached at least 90% accuracy.

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

Three systems in particular worked best at TempEval-2, though their strengths lie in different places. HeidelTime (Strötgen and Gertz, 2010) is a modular rule-based system including a large ruleset; this enabled it to achieve top performance at timex normalisation. However, rule-based approaches are likely to face diminishing returns as they attempt to raise recall through introduction of new rules. TRIPS/TRIOS (UzZaman and Allen, 2010) and TIPSem-B (Llorens et al., 2010) are both systems that use machine learning for timex recognition, with sophisticated feature sets. Using the TempEval-2 data released after the exercise, it has been shown to be possible for very simple feature sets to reproduce state-of-the-art timex recognition performance (Llorens et al., 2011). Normalisation remains a task that appears to requires a rule-driven solution, with promising new systems emerging (Llorens et al., 2012a).

2.4 Chapter Summary

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

Chapter 3

Temporal Relations

> L'étude a été pour moi le souverain remède contre les dégoûts, n'ayant jamais eu de chagrin qu'une heure de lecture ne m'ait ôté.
> *I found study was always the best remedy for worries, having never known any trouble that an hour's reading did not assuage.*
>
> MONTESQUIEU

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.

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 an exhaustive 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, (Setzer and Gaizauskas, 2001; Setzer, 2001)), TimeML (Section 2.3.2) defines its own set of interval relations, based on Allen's interval algebra (Allen, 1983); 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 Section 3.2.

For the context of this thesis, 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 (Hobbs and Pan, 2004).

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 (Prior, 1968) and Bruce's model (Bruce, 1972), 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 (Moens and Steedman, 1988; Goranko et al., 2004; Denis and Muller, 2010).

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 present in natural language discourse.

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, Setzer et al. (2005) 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.

Temporal Interval logic

Allen's interval logic (Allen, 1983) 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

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

Table 3.1: Allen's temporal interval relations

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 B.9.

A story typically describes more than one event, with some temporal ordering. Example 3.1 describes two events, setting out (E1) and living happily (E2).

(3.1) *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 3.2.

(3.2) *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

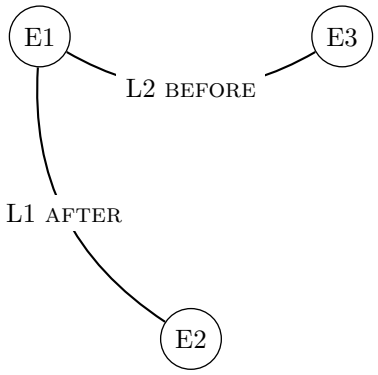

Figure 3.1: Temporal graph of a simple story.

- 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 Figure 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. McDermott (1982), 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 Goranko et al. (2004) and Galton (2008).

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 (Vilain and Kautz, 1986; Tsang, 1987). Problems arise when dealing with instantaneous events (e.g. "improper" intervals – Section 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 Allen and Hayes (1989). 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.

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 3.3 contains an excerpt from a news report, uttered mid-way through a day. The timex *Today* has a specific meaning of a 24-hour 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*.

(3.3) *Today, rebels still control the airfield and surrounding area.*

To this end, Freksa (1992) 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 (Kowalski and Sergot, 1989). 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, Section 6.4.2 investigates semi-interval logic in the context of tense-based temporal relation typing.

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

Table 3.2: TimeML temporal relations

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 (Verhagen, 2004) and CAVaT (Derczynski and Gaizauskas, 2010a) exploit. They also better lend themselves to graph-based reasoning about temporal structures in text (Denis and Muller, 2011). 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 (Goranko et al., 2004).

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.

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

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

3.3 Temporal Relation Annotation

The work in this thesis primarily concerns temporal relation annotation using intervals, as opposed to points or semi-intervals.

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. Thereofre, 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 & 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 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

[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.

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 Mani et al. (2006)) and there are mappings to the simple SIMULTANEOUS/BEFORE/INCLUDES relations specified by Setzer et al. (2005). 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 fact removes any influence that annotator preference may have and presents data in a uniform manner.

Based on Table 1 from Mani et al. (2006), 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 Setzer et al. (2005), from Vilain and Kautz (1986), 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 (Pustejovsky, 2009), 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 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.

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

Table 3.4: Relation folding mappings used in this thesis.

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 3.5: Distribution of TLINK relation types in TimeBank 1.2

	Unclosed		Closed	
Relationship type	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	

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

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 thesis 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, de-prioritising other more tenuous or remote links (Setzer and Gaizauskas, 2000). 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 possible 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 Figure 3.1. Graph-based representations lead to sophisticated reasoning (Denis and Muller, 2011) and evaluation measures (Section 3.4.4).

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 Mani et al. (2007) (see Section 3.4.1). We fully investigate comparison of temporal annotations in Section 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 thesis 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.

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 B.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 (Section 3.4). After this, a set of previous approaches are described, in terms of their dataset, features and performance (Section 3.5). The progress in the field so far is then summarised and an analysis presented (Section 3.5.6).

3.4.1 Closure for training data

In order to provide extra training data, temporal closure (Verhagen, 2005) 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 Section 3.3.2

3.4.2 Global constraints

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

(3.4) A BEFORE B

(3.5) 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, Mani et al. (2007) propose only allowing document-level splits in data.

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 (Chklovski and Pantel, 2004b,a). The data comes from mining Google results using templates (Lin and Pantel, 2002) 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 (Chambers and Jurafsky, 2008b) 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 3.6.

(3.6) *The president's son* $\underline{met}_e$ *with Sununu* $\underline{last\ week}_t$.

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 (Setzer et al., 2005; Boguraev et al., 2007). 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 (Verhagen et al., 2007). Humans, for example, have trouble distinguishing some relations such as IS_INCLUDED and DURING (Bethard et al., 2007b). 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.

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 (Setzer et al., 2005; Verhagen, 2005; Tannier and Müller, 2011) 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 (Mani et al., 2007).

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.

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 Section 3.2). Temporal closure of the graph can be computed, leading to a more consistent representation of the annotated data (Allen, 1983; Vilain and Kautz, 1986; Setzer et al., 2005). It is possible to measure agreement between graphs (Verhagen et al., 2007).

Not all relations have the same importance; some entail more information – some may lie on something akin to a critical path (Kelley Jr and Walker, 1959), 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. Tannier and Müller (2011) propose a method for this, involving 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 (Levenshtein, 1965).

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 Figure 16 of (Tannier and Müller, 2011). 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 it are difficult to compare to previous work. This should be remedied somewhat in TempEval-3 UzZaman et al. (2012), which uses a TimeGraphs-based temporal evaluation metric UzZaman and Allen (2011).

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 (Verhagen et al., 2009). TempEval and TempEval-2 both use a simplified set of relations and a purpose-created corpus. Systems in TempEval-2 (Verhagen et al., 2010) 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 (UzZaman et al., 2012) also requires participants to perform temporal relation identification.

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 Section 5.6.3.

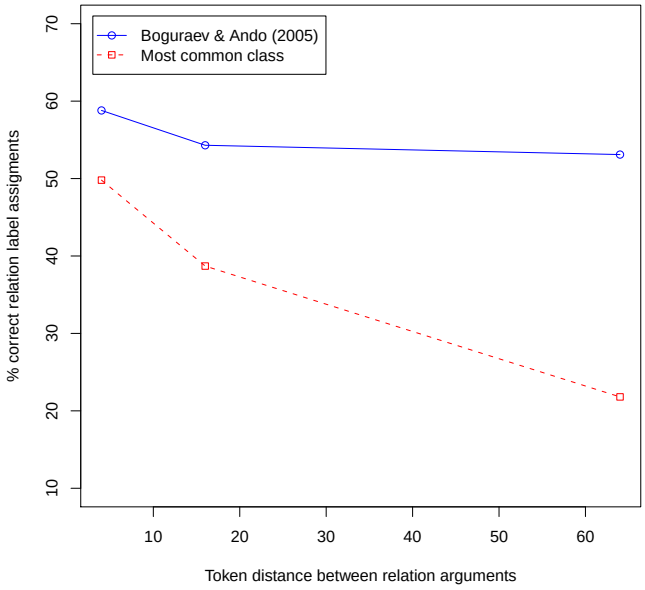

Figure 3.2: TLINK relation type assignment difficulty increases with the distance between link arguments.

One of the earliest approaches Boguraev and Ando (2005), 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 (Zhang et al., 2002) 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 thesis), 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 Section 2.3.3) to the feature set consistently provides a small absolute performance boost (0.7% to 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 Figure 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 Boguraev and Ando (2005)'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, Hepple et al. (2007) 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, (Mani et al., 2006) 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 Mani et al. (2007) 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 Section 3.3.2).

The corpus used is a merging of a custom version of TimeBank (Pustejovsky et al., 2003) (v1.2a – not publicly available) and the Aquaint TimeML Corpus (ATC) (ARDA, 2006). 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 (Mani et al., 2007) – that is, using a

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

combined TimeBank/AQUAINT corpus plus the TimeML element attributes as features. Using support vector machines, Mirroshandel et al. (2010) 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 Mani et al. (2007)) and 67.03% for event-event relations on the same (compared to 70.4% from Chambers and Jurafsky (2008a), detailed below).

Vasilakopoulos and Black (2005) 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 competitive 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 Mani et al. (2003), 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 (Reichenbach, 1947). 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. (2011) 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 (Minsky, 1991). 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. (2010a) 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. Kolya et al. (2010b) take a similar approach, 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 Mani et al. (2007), Min et al. (2007) 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 (Hagège and Tannier, 2007), 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 (Yoshikawa et al., 2009).

Syntactic relations can also play a role in determining temporal relation types. Bethard et al. (2007a) 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 (Verhagen et al., 2007), 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, Bethard et al. (2007b) created a custom corpus of verb-clause event pairs, using TANGO (see Section B.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 Boguraev and Ando (2005).

Cheng et al. (2007) 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, UzZaman and Allen (2010) 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, Marsic (2011) built on their earlier approach (Puşcaşu, 2007b) 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 et al. (2010) 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 Llorens et al. (2010). The concepts are further explored in Llorens et al. (2012b), 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.

VerbOcean is a resource detailing semantic relations between verbs, mined from large corpora. One of these relation types is temporal: *"happens-before"*. Mani et al. (2006)'s system includes experiments which perform VerbOcean (Section 3.4.2) 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 (Bethard et al. (2007a); Derczynski and Gaizauskas (2010c)). However, when the features that help in these specific situations are applied generally, they lead to a performance

[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." (Mani et al., 2006)

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.

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 (Verhagen et al., 2009, 2010) 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 Chapter 5.

As part of a TempEval system, Derczynski and Gaizauskas (2010b) 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 (Derczynski and Gaizauskas, 2010c), 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 (Chapter 5).

Finally, Tatu and Srikanth (2008) experiment with the addition of event participant and event co-reference 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 (2008a) manually add links to TimeBank v1.2 in cases where events subordinate other events in the same clause (as per Bethard et al. (2007a)) 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, Yoshikawa et al. (2009) 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 (Dang et al., 2008), forensic analysis (Howald and Katz, 2011) or temporal slot filling (Ji et al., 2011; Burman et al., 2011).

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.

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 (Chambers et al., 2007) or after attempts with a sophisticated array of cutting-edge classifier kernels (Mirroshandel and Ghassem-Sani, 2011; Mirroshandel et al., 2011). Even the introduction of some syntactic information such as argument ancestor path distance and is not sufficient to overcome this barrier (Mirroshandel et al., 2010; Mirroshandel and Ghassem-Sani, 2010). Taking care of other information sources, such as global constraints, yields an immediate but small performance increase over the base feature set (Chambers and Jurafsky, 2008a; Yoshikawa et al., 2009).

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 Section 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.

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 Hovy et al. (2006). 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. Freksa (1992)). 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.

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

System	Notes	Method	E-E	E-T
Lapata and Lascarides (2006)	BLLIP corpus	decision tree	70.7	
Gaizauskas et al. (2006)	Clinical corpus	rule-based	65	
Bramsen et al. (2006)	Medical discharge summaries	graph based	78.3	
TempEval-1 corpus				
Baseline	*Most common class*		*47*	*57*
Cheng et al. (2007)	Uses dependency parsing	HMM SVM	49	61
Hepple et al. (2007)	Includes text order features	SVM / K*	54	59
Bethard and Martin (2007)	Uses syntactic tree features	SVM	54	61
Marsic (2011)		rule-based		65
Kolya et al. (2011)		CRF + rules		75.9
Puşcaşu (2007b)	Syntactico-semantic features	rule-based	54	**80**
Min et al. (2007)	Focus on rules for marginal cases	SVM	55	58
Kolya et al. (2010a)		CRF	55.1	73.8
Hagège and Tannier (2007)	Based on XIP deep parse data	rule-based	**57**	34
Yoshikawa et al. (2009)	Models global TLINK constraints	MLN	**57**	65
Bethard et al. (2007b)	Same-sentence links only	SVM + rules	89.2	
TempEval-2 corpus				
Baseline	*Most common class*		*48.63*	*55.07*
Derczynski et al. (2010b)	Includes signal information	MaxEnt + rules	45	63
Ha et al. (2010)	Lexico-syntactic feat. + VerbOcean	MLN	51	63
Llorens et al. (2010)	Includes semantic features	CRF	55	55
Kolya et al. (2010b)	Includes event head information	CRF	56	63
UzZaman and Allen (2010)	Based on TRIPS parse data	MLN	58	**65**
TimeBank 1.1 corpus				
Baseline	*Most common class*		*33.38*	
Boguraev and Ando (2005)	Token windows, FS-grammar features	RRM		53.1
Vasilakopoulos and Black (2005)	Not using folded relations	K*	53.14	
Chambers et al. (2007)	Segregates intra-sent. relations	SVM	**67.57**	
TimeBank 1.2 corpus				
Baseline	*Most common class*		*38.35*	*58.4*
Puşcaşu (2007a)	Maps to TempEval relations	rule-based	53	65
Tatu and Srikanth (2008)	With actor and co-ref features	SVM	58.2	
Mirroshandel, Ghassem-Sani (2010)	Bootstrapped kernel w/ AAPD	SVM	66.18	
Chambers and Jurafsky (2008a)	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 et al. (2007)	Uses TimeBank 1.2a	MaxEnt	(59.68)	(82.47)
Mirroshandel et al. (2010)	LICT Polynomical kernel	SVM	67.03	**80.04**
Mirroshandel, Ghassem-Sani (2010)	Bootstrapped kernel w/ AAPD	SVM	68.07	
Derczynski et al. (2010c)	Signalled TLINKs only	MaxEnt	82.19	

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.

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 and Katz (2009) 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.

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 body of this thesis.

Chapter 4

Relation labelling failure analysis

Felix, qui potuit rerum cognoscere causas
Fortunate was he, who was able to determine the causes of things

Georgica (II, v.490)
Virgil

4.1 Introduction

In Chapter 3, we discovered that automatic temporal relation typing is a difficult problem. This motivates an investigation into potential ways of improving performance in relation typing. This chapter details an attempt to discover potential ways of improving performance at the task. As humans are readily able to identify the nature of temporal links, one may *a priori* draw the conclusion that the information required to do so must be available somewhere. This knowledge is in a given document or in information known by the reader before encountering that document (referred to as **world knowledge**). We attempt to characterise and enumerate the in-document knowledge used to support temporal link labelling. In later chapters, we will use some of these types of knowledge to improve automatic temporal relation labelling.

Firstly, this chapter reports on an attempt to identify a common set of challenging temporal links in the TempEval-2 evaluation task. This includes re-examination of the surface information available in TempEval-2 data and an analysis of its distribution in difficult links. Secondly, finding that the surface information presents no clear paths for investigation (as suggested by the performance cap of previous work using surface information discussed in Section 3.5.6), a manual investigation of difficult links is undertaken. This comprises a qualitative characterisation of the information used to label the links and motivates our later experimental investigations.

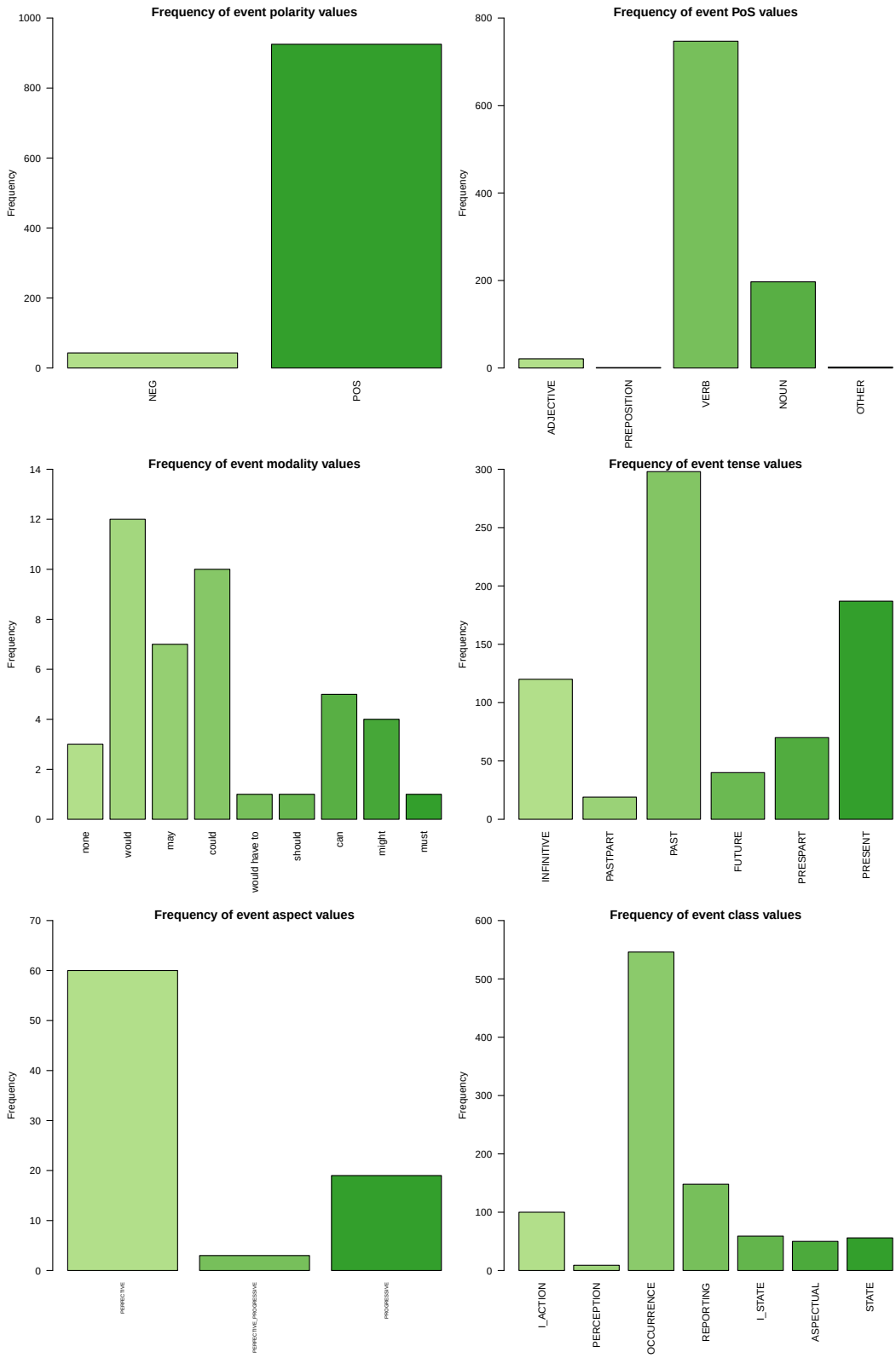

Figure 4.1: Frequencies of event attribute values in the TempEval-2 English test data.

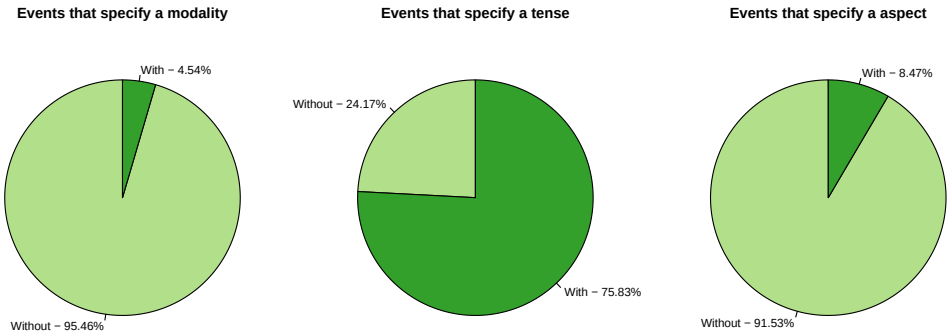

Figure 4.2: Proportions missing events attribute values in the TempEval-2 English test data.

4.2 Survey of difficult TLINKs

Our hypothesis is that there may be temporal relations that are consistently difficult to classify correctly. That is, some meta-system using an agglomerative approach (e.g. voting) will still have problems with the relation typing problem. It has been difficult to conduct a thorough error analysis of the temporal relation typing task, as authors often do not or cannot make their attempted labellings available, instead publishing more concise overall performance figures. Further, there are many different corpora and corpora-versions used, which hampers comparability.

This section introduces a source of data on attempts at the relation labelling task, followed by a method for grading temporal links in terms of difficulty, reports on the measured proportions of the degrees of difficulty found in typing various temporal relations, defines what constitutes a difficult link and finally presents a data-driven analysis of difficult links based on their surface features.

4.2.1 The TempEval participant dataset

As mentioned in Section 3.4.4, the TempEval exercises strive to produce comparable results over a fixed and agreed dataset, using pre-annotated events, timexes and TLINK arguments, which constrains the scope for variation in systems outside the task focus – temporal labelling methods.

The second TempEval exercise took place in 2010, as part of SemEval (Verhagen et al., 2010). This exercise included four temporal link labelling exercises, in multiple languages, over a purpose-built corpus. Many teams participated in the evaluation and attempted to label these temporal links. As a result, from their submissions we gained a snapshot of the state of the art of temporal link labelling, all on the same data, with multiple approaches. Some teams were prepared to share their submitted results, which, when compared with the correct answer data and the original corpus, could be merged. From this, we were able to measure a "success rate" for each temporal link, determined by the proportion of systems that managed to label it correctly. We then can build a list of links that are difficult for most (or all) of the systems to annotate automatically.

Fortunately, the TempEval-2 organisers released a full dataset of not only source but also eval-

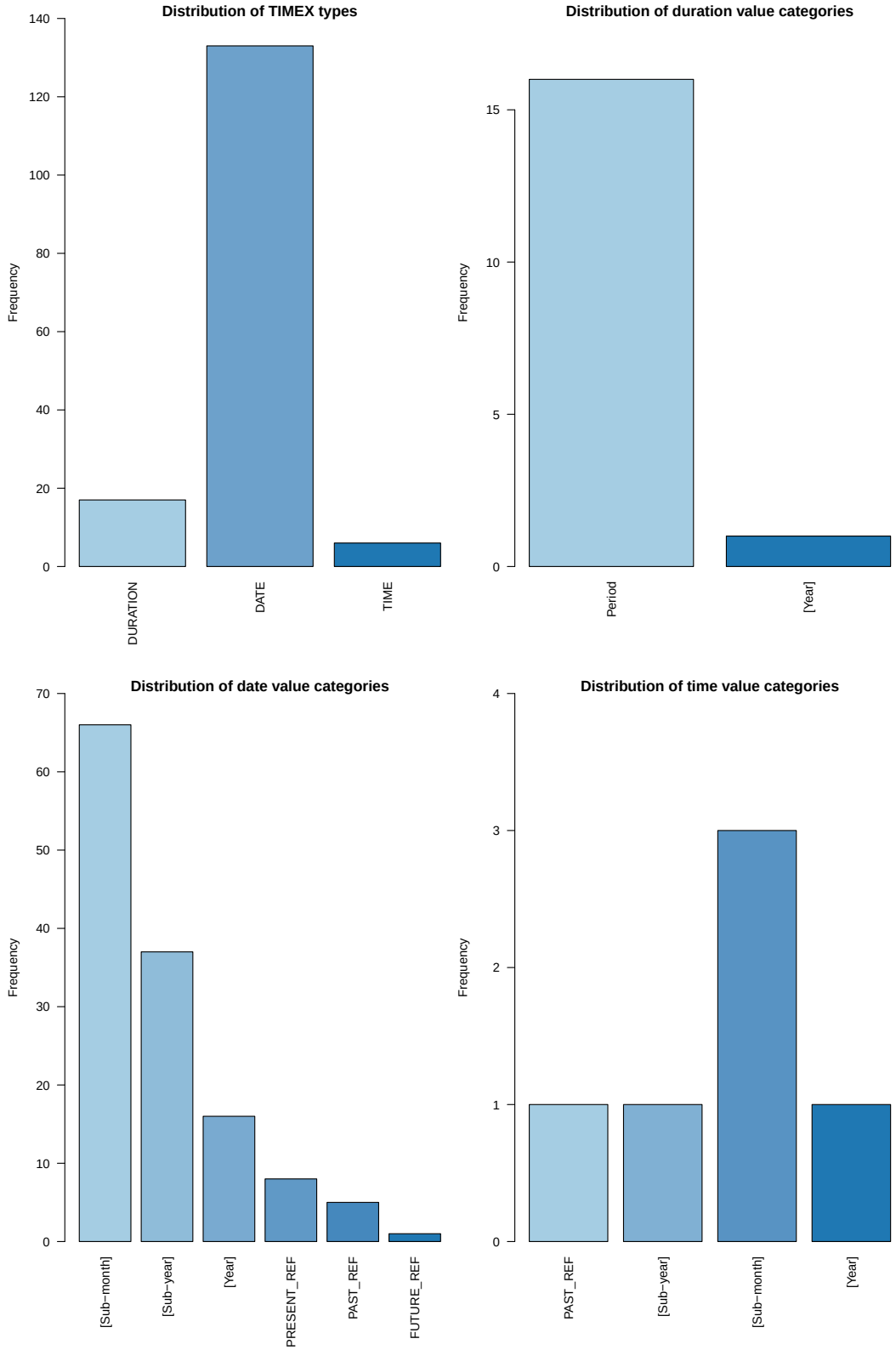

Figure 4.3: Frequencies of timex attribute values in the TempEval-2 English test data.

Task	Difficult links	Difficult proportion	Best score
C	22	8.59%	65%
D	39	18.4%	82%
E	62	44.3%	58%
F	44	46.8%	66%

Table 4.1: Proportion of difficult links in each TempEval-2 task

uation data.[1]. Data concerning the distribution of features over events are contained in Figures 4.1 and 4.2, of features over timexes in Figure 4.3.

After contacting teams participating in temporal relation labelling tasks, many were kind enough to donate their submitted labels (Ha et al., 2010; Derczynski and Gaizauskas, 2010b; UzZaman and Allen, 2010; Llorens et al., 2010). This data was used to conduct a data-driven failure analysis of four separate temporal linking tasks undertaken by directly comparable systems. The analysis continues the work on TempEval-1 by Lee and Katz (2009) and incorporates data from many individual teams.

Given the apparent performance ceiling of systems that use only the annotated TimeML / TempEval-2 feature:value pairs (surface information), clear directions for further investigation are not expected from a formal analysis using these feature:value pairs. However to omit an analysis of difficult links in terms of their arguments' TempEval-2 descriptions would be to ignore a potentially useful and readily available information source and so results are included below.

4.2.2 Defining what constitutes "difficult" temporal links

We start by measuring the "difficulty" of each link, calculating the proportion of attempting labelling systems that generated a correct response. The measurements have values ranging from "all systems correct" (an easy link) to "no systems correct" (a difficult link). This gives a discrete set of difficulty categories for each task. We then count the number of links in each difficulty category as a proportion of the whole and present the data graphically. The results are shown in Figure 4.4.

- **Task C** – Linking events and timexes in the same sentence. For example, *The day$_t$ before Raymond Roth was pulled$_e$ over ....*
- **Task D** – Linking events with the document creation time. For example, *11/01/89$_t$. ... As part of the agreement, Cilcorp said$_e$ it will co-operate..*
- **Task E** – Linking main events in adjacent sentences. For example, *There are 12 flood warnings in the South West, with Met Office warnings for snow covering$_{e1}$ much of the UK. This comes$_{e2}$ just over a week before the start of British Summer Time..*
- **Task F** – Linking main events with subordinate events. For example, *He said$_{e1}$ he discussed$_{e2}$ the issue with Mr. Netanyahu..*

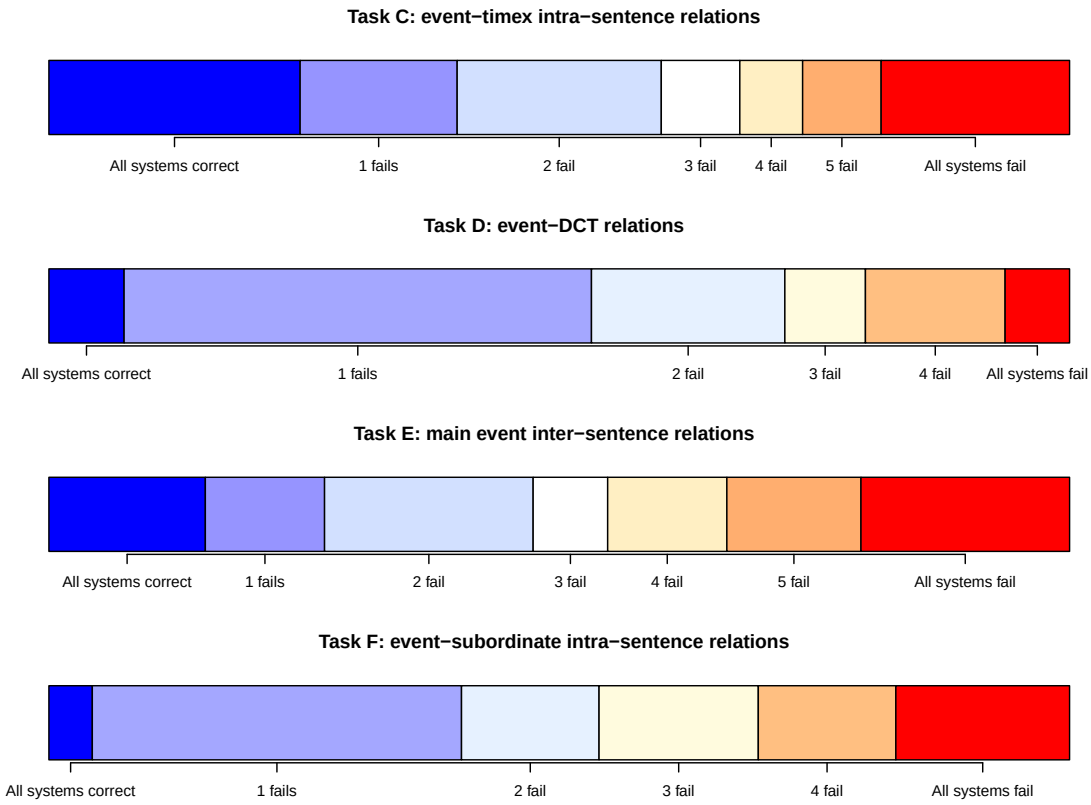

Figure 4.4: TempEval-2 relation labelling tasks, showing the proportion of relations organised by number of systems that failed to label them correctly. Six systems attempted tasks C and E; five attempted tasks D and F.

Systems in error	Number of TLINKs	% of TLINKs
No faults	16	24.6%
1 fault	10	15.4%
2 faults	13	20.0%
3 faults	5	7.69%
4 faults	4	6.15%
5 faults	5	7.69%
All fail	12	18.5%

Table 4.2: Error rates in TempEval-2 Task C, event-timex linking

Systems in error	Number of TLINKs	% of TLINKs
No faults	14	7.37%
1 fault	87	45.8%
2 faults	36	18.9%
3 faults	15	15.8%
4 faults	26	21.1%
All fail	12	6.32%

Table 4.3: Error rates in TempEval-2 Task D, event-DCT linking

This information permits a brief overall analysis of the relative complexity of the different relation tasks. Task E (Table 4.4) has a fairly stable difficulty gradient, with the least deviation between category sizes. Task D (Table 4.3) is easiest. Task C (Table 4.2) has a very tough set; when compared to task E (Table 4.4), although a greater proportion of the links are successfully labelled, the size of the "all fail" group is the same in absolute terms and relatively dominates the set of harder links. Finally, it can be seen that event-event labelling (tasks E+F, Tables 4.4

[1]Downloadable from http://timeml.org/site/timebank/tempeval/tempeval2-data.zip It is important to note that this contains more data than was in the tasks set; evaluating systems using this release as-is will not give accurate figures.

Systems in error	Number of TLINKs	% of TLINKs
No faults	21	15.3%
1 fault	16	11.7%
2 faults	28	20.4%
3 faults	10	7.30%
4 faults	16	11.7%
5 faults	18	13.1%
All fail	28	20.4%

Table 4.4: Error rates in TempEval-2 Task E, linking main events of subsequent sentences

Systems in error	Number of TLINKs	% of TLINKs
No faults	6	4.26%
1 fault	51	36.2%
2 faults	19	13.5%
3 faults	22	16.1%
4 faults	19	13.5%
All fail	24	17.5%

Table 4.5: Error rates in TempEval-2 Task F, linking events to events that they subordinate

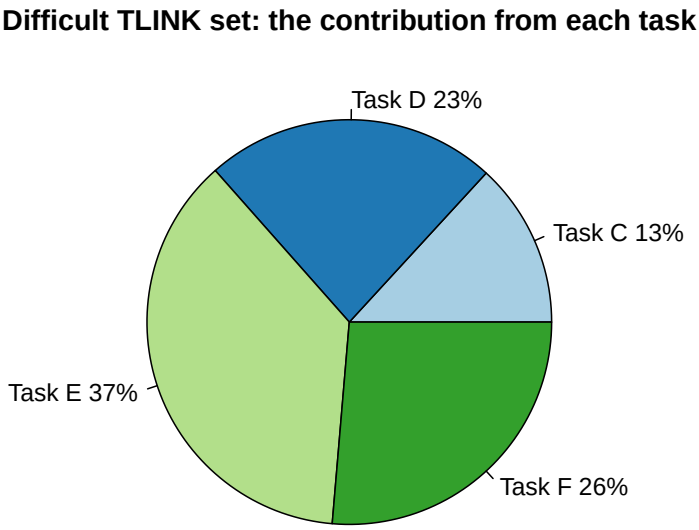

Figure 4.5: Composition of the set of difficult links. Event-event tasks (E and F) in green, event-timex tasks (C and D) in blue.

and 4.5) is harder than event-timex labelling (C+D, Tables 4.2 and 4.3).

Data was available for five or six systems, depending on the task. One system only attempted two of the four tasks, so its absence should not unduly undermine the quality of overall observations. Difficult links are defined as those wrongly labelled by all systems or wrongly labelled by all-but-one system. Given this threshold, we can define a set of difficult links for further analysis. The composition of this set is given in Table 4.1 and shown in Figure 4.5.

Figure 4.6 shows the proportion of links within each task that are difficult and reinforces the earlier observation that event-event links are tougher than event-times links. In the figures, event-timex tasks (C and D) are shown in blue and event-event tasks (E and F) in green. Event-event tasks are comparatively hard, with higher proportions of difficult TLINKs.

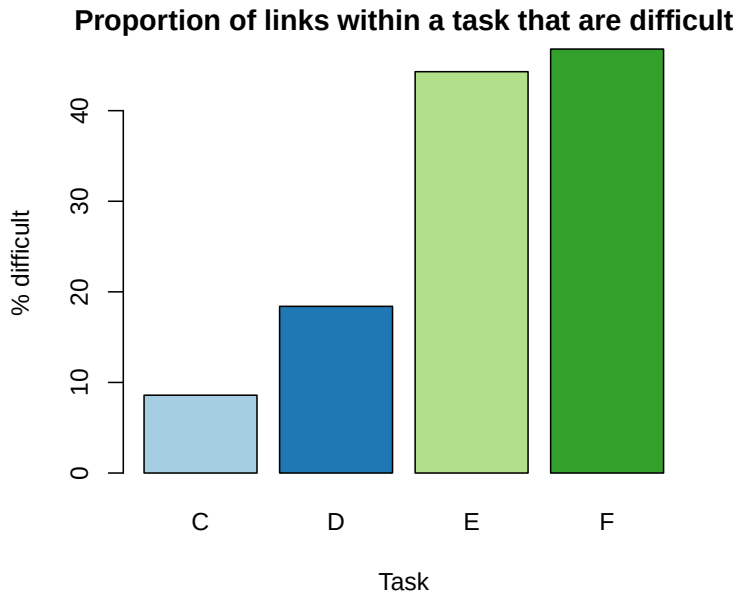

Figure 4.6: Proportion of each TempEval-2 task's links that are difficult.

4.2.3 Comparative distribution analysis

Given a set of gold-standard event annotations and gold-standard temporal link annotations, one can conduct a survey of features and values for temporal links. Given also a set of difficult links, one may determine which particular attribute combinations are difficult or easy to automatically label. This is demonstrated in Figure 4.7, which may be read as follows. Each row corresponds to all events *related to* a given event having a particular property. For example, one row may detail the statistical properties of all other events that are linked to a verb event (e.g. having `pos.VERB`). The columns in this row show the distribution of feature/value pairs in the related event for all relations surveyed. So, continuing the example, in the `pos.VERB` row, the colour represents the likelihood of other argument in the temporal link having a particular feature/value pair. More saturated colours represent higher frequencies. Reds indicate relatively high presence in difficult links (e.g., a "hard" feature combination); blues indicate a low frequency in difficult links (e.g., that the feature combination is "easy").

One could imagine that graph 1 minus graph 2 is graph 3 and that the reds correspond to negative values. Let $\mathbf{A}$ be a matrix of feature:value co-distributions and $\mathbf{B}$ be feature co-distributions in the set of difficult links. If comparison $\mathbf{O} = \mathbf{A} - \mathbf{B}$, then negative values in $\mathbf{O}$ correspond to feature combinations that occur more frequently in $\mathbf{B}$ than $\mathbf{A}$; that is, combinations that are more likely than average to be occur in difficult relations.

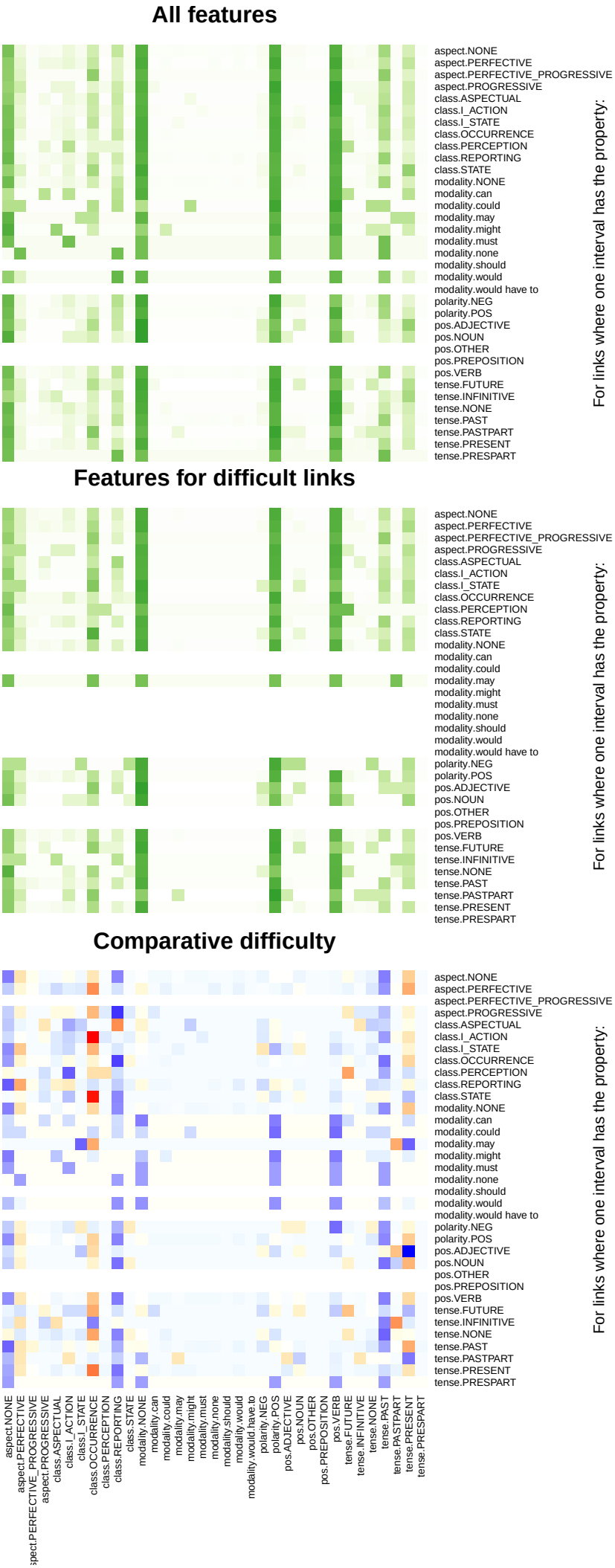

Figure 4.7: Comparative analysis of features for TempEval-2 task E.

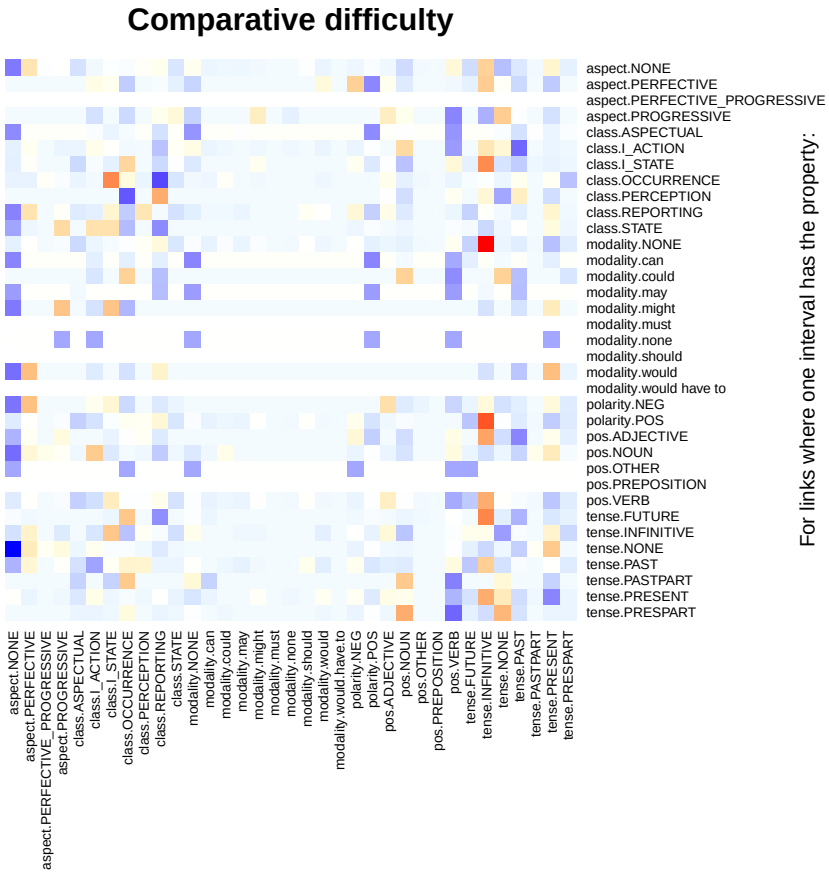

Figure 4.8: Comparative analysis of features for task F, relating events with their subordinate events.

Difficult Event-event link attribute distribution

Following this, the figure (4.7) presents three saturation maps. The first shows the feature:value co-distribution matrix for all relations. The second shows the matrix just for the difficult relations in that task. By subtracting the second from the first, we can derive the difference between all relations' feature:value distribution and just the difficult relation's distributions. That is, we can identify feature:value pairings that are easier or harder to classify. The harder examples are in red, the easier in blue. Where the distribution varies little between all links and just difficult links, the tone tends to white (unsaturated). Thus, a red cell (for example, where an event of `class.I_STATE` is related to a different event which has `aspect.PERFECTIVE`) represents a frequently difficult combination. Conversely, a dark blue cell (e.g. when an adjective is linked with a present-tense event) shows an easy combination; that is, a pairing which, though frequent, is rarely found in the difficult set. The graphs should not exhibit symmetry, because each row represents a different prior assertion, and is the distribution of other features given that assertion, whereas columns do not represent priors.

This information for Task E, linking main events in successive sentences, is in Figure 4.7, and

for Task F, that of linking events where on linguistically subordinates the other, is presented in Figure 4.8.

For Task E, from the vertical red stripe in the differential diagram, it can be seen that links to *occurrence*-class events were particularly difficult to label, especially when the other event is of class state or intentional action. However, links to *reporting*-class events were generally easier than average. This could perhaps be due to better consistency in annotations leading to better supervised models, or that a reporting event is typically after the events that are reported but before DCT, giving inherent constraints to this event class. Aside from links with reporting events, particularly easy were links between perceptions and intensional actions (perhaps with perceptions encouraging a reaction?) and links between adjectives and present-tense verbs (perhaps because these always overlap – e.g. *"He says it's hot out there."*).

As for Task F (Figure 4.8), links with verbs that have no aspect seem to be consistently easier than most. There is less variation in difficulty between certain feature pairings when compared to Task E, as evidenced by the comparatively less saturated graph. Links to infinitive or un-tensed arguments (e.g. non-verbs) seem to present more difficult than other parts of speech. Of note for being difficult are cases where there is no modality specified in one event and the other is infinitive, possibly due to a reduced number of amodal training examples in a set dedicated to subordination; with links between an occurrence and a state; and with links between future-tense verbs and infinitive verbs.

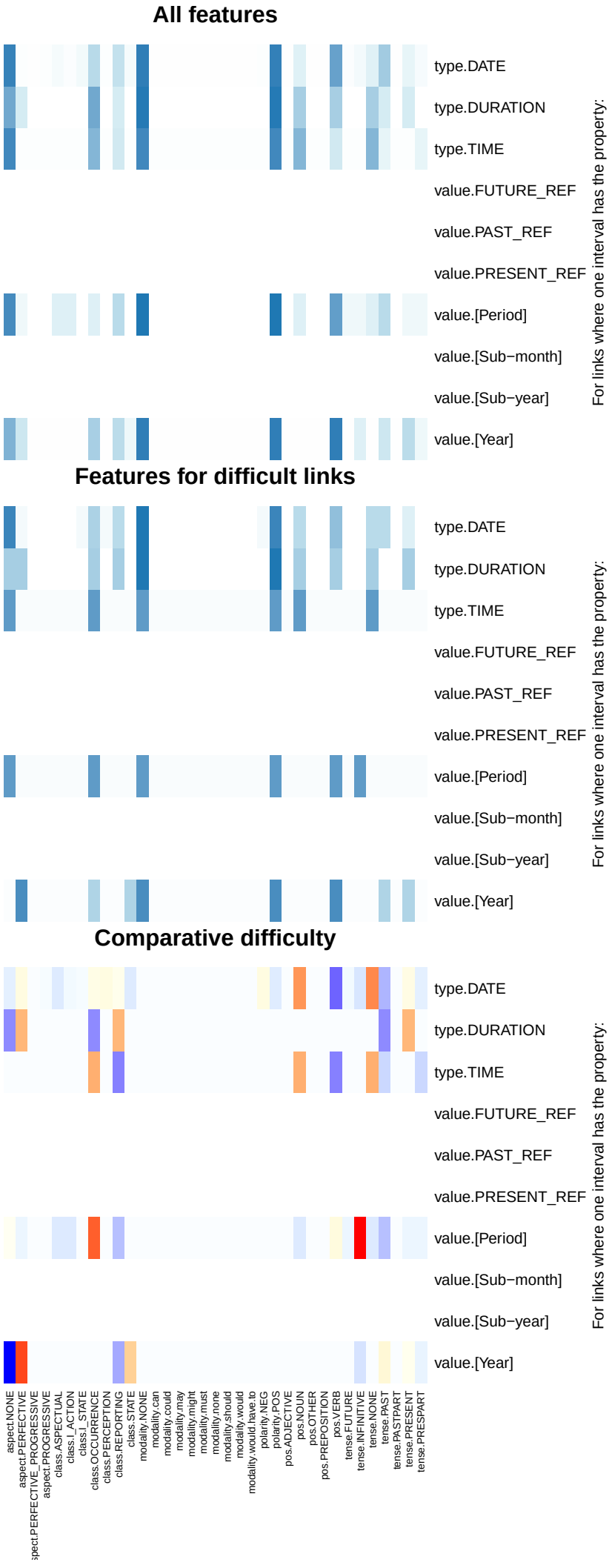

Figure 4.9: Comparative analysis of features for TempEval-2 task C.

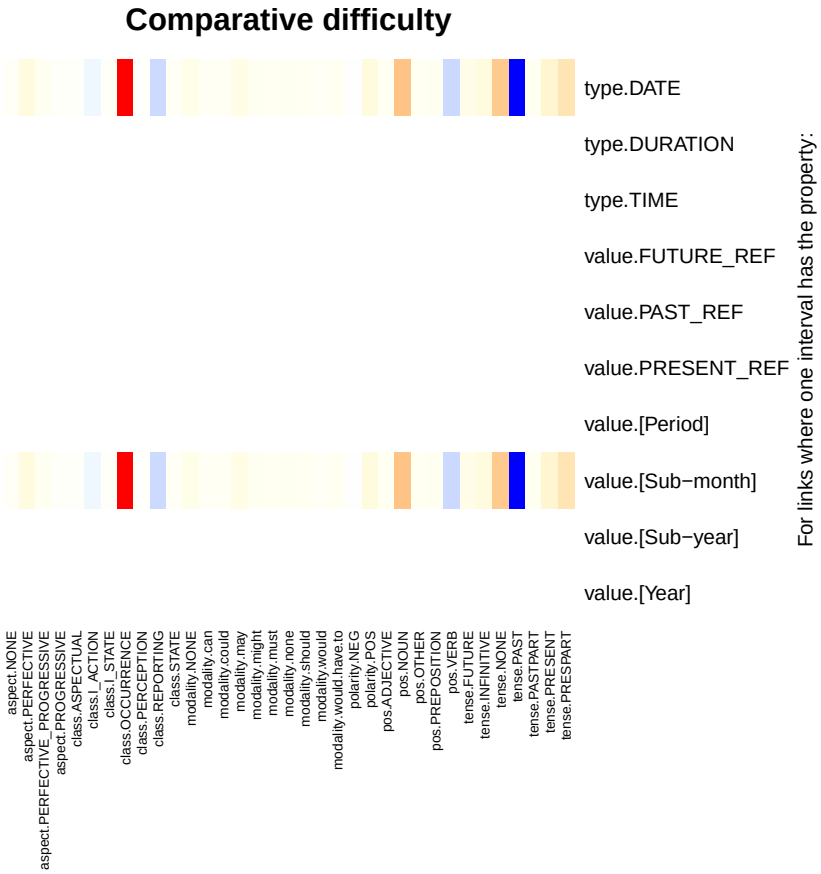

Figure 4.10: Comparative analysis of features for task D, relating events to DCT.

Difficult Event-timex link attribute distribution

The corresponding data for Tasks C and D are shown in Figures 4.9 and 4.10 respectively. The colour scheme for event data in green and timex data in blue is continued here, with the exception of comparative difficulty graphs, which use a red/blue divergence colour scheme. In these cases, deep reds indicate very difficult combinations and blue blues very easy ones. Note that the data for task D is only for date-type timexes of granularity less than a month, because in all cases the timex refers to a specific date – DCT – in the data.

For Task C, times, dates and duration appear to be difficult with different sets of event features. Dates and times are difficult to relate correctly to nouns, whereas durations are heard to link to occurrences and present tense verbs. Interestingly, year-sized timexes are very difficult to correctly link to progressive verbs, but very easy to relate to events with no aspect information.

In Task D, we do not have much information. This may be due to a small number of timexes being present in this task's difficult set; the task turned out to be relatively easy. Of these, they are easier to relate correctly to past tensed verbs, and harder to link to occurrence-type events.

4.2.4 Attribute distribution summary

It was consistently found that temporal relations between two events are harder to classify than relations between an event and a time. This should direct future research efforts, and was the focus of the latter part of the section, which related a more detailed investigation into the properties of the intervals coupled in difficult links.

Regarding patterns in attribute values over difficult links, although some specific situations of high frequency of difficult links are identified, no clear overall picture emerges. A few specific cases were identified as consistently difficult or easy, but these generally comprised a small proportion of all links. For example, perfect aspect events were had to relate to timexes lasting a year or more; occurrence-class events were difficult to relate with other events, and reporting-class events were easier to relate with other events; and adjective events were easy to relate to present-tense events.

We lead in the next section to a more qualitative approach, taking phenomena contained elsewhere in annotations or not in annotations at all and examining their prevalence in difficult links.

4.3 Extra-feature analysis

The overall goal is to determine linguistic sources of temporal ordering information. Because the annotated features do not appear to contain enough information to automatically label links (Section 4.2, Chapter 3), other sources of information must be considered. Formal analysis of the surface data does not present immediate clues. This section presents the results of a survey of each link in the TempEval-2 "difficult set" in terms of the type(s) of information required to determine the temporal relation, aside from that given in TimeML annotations. The resulting information is then used in the next section to attempt to characterise information that temporal links may draw upon, based on prior knowledge about linguistic representations of time.

This analysis was conducted independently of available models and tools, focusing instead on linguistic phenomena. This is to reduce bias from existing methods for and knowledge of the problem. To this end, no TimeML annotation features, tense models or linguistic processing tools were used to construct criteria for characterisation.

4.3.1 Characterisation

It is useful to analyse the difficult TLINKs in a manner that allows identification of common traits. While one can qualitatively express what information is used express a temporal ordering in discourse, to feed into a computational approach one requires quantifiable or at least discrete measures that can be taken consistently from all links. To this end, a set of readily-identifiable linguistic phenomena were determined that could provide temporal information beyond those expressable in TimeML. Each difficult TLINK is then examined and a record made of whether or not each of these phenomena is in place. The result is a survey of types of information used to support temporal orderings for the set of TempEval-2 difficult TLINKs.

	Task			
Description	**C**	**D**	**E**	**F**
Total instances	21	38	62	43
Signalled	33.33%	13.16%	11.29%	6.98%
Inference	61.90%	42.11%	30.65%	9.30%
World knowledge	9.52%	2.63%	14.52%	9.30%
Iconicity	19.05%	0.00%	37.10%	34.88%
Unclear/Disagree	14.29%	18.42%	4.84%	4.65%
Same sentence	100.00%	0.00%	0.00%	97.67%
Same clause	19.05%	0.00%	0.00%	30.23%
Tense shift	0.00%	0.00%	37.10%	34.88%
Differing modalities	47.62%	34.21%	8.06%	51.16%
Differing progression	0.00%	0.00%	16.13%	11.63%
Causal	0.00%	0.00%	9.68%	4.65%

Table 4.6: Temporal ordering phenomena and their occurrence in difficult links.

The set of phenomena is listed below. Each link may use any number of phenomena. The set is broken into two types: information about the relation and the ordering and information about the interaction between arguments in text.

Relation information

- **Signalled** - the relation intervals is explicitly expressed by a co-ordinating temporal conjuction or phrase (such as *before*).
- **Inference** - the relation can be easily inferred by reasoning involving other relations in the document
- **From world knowledge** - external information about the general structure of complex events can help determine this relation
- **Iconicity** - temporal order of relation arguments matches the order of their appearance in the source text
- **Disagree** - the annotated relation type is in dispute

Arguments in text

- **Same sentence** - the relation's arguments are in the same sentence
- **Same clause** - the relation's arguments are in the same clause
- **Tense shift** - there is a shift of tense from one argument to the other
- **Differing modalities** - the arguments do not have the same modality or are not in the same conditional world
- **Differing progression** - one argument is progressive or signifies a culmination or has another aspectual difference from the other
- **Causal** - one argument causes the other and this is critical to the ordering

A "world knowledge" category is therefore included in the above list, in an attempt to roughly estimate how often extra-discourse information is required to resolve difficult links. Also, a "not clear" category is present, for cases where one disagrees with the gold standard.

4.3.2 Analysis

The proportion of difficult links that use each of these phenomena as part of their temporal ordering information is shown in Table 4.6.

Overall, 11.2% of all TLINKs in TimeBank are annotated as using an explicit temporal signal. It seems that a greater-than-average proportion of difficult intra-sentence event-time links rely on signals (task C), but that difficult subordinated relations (task F) use them less often than is typical.

World knowledge rarely supported difficult links. The task that it helped in most was linking main events in adjacent sentences.

Iconicity – that is, when temporal order follows discourse mention order – was generally not observed within the difficult links set. No task had more than 40% of its difficult links in the same textual and temporal order. The prevalence of iconicity was higher in difficult event-event links than event-timex. This may be because it is somewhat redundant in the case of `DATE` and `TIME` timexes, because the timex provides an explicit temporal reference point, and one has less need to rely on implicit factors in order to situate link arguments. Nevertheless, it is interesting to observe that times earlier than events tended to be mentioned in text *after* the events, for the difficult link set. It may also be the case that general discourse follows the principle of iconicity Diessel (2008) and that having made this observation, automatic temporal relation systems run into difficulties when the principle does not apply.

For event-event links (tasks E and F), a notable proportion of difficult links employ a tense shift. This is where the tense dominating one event is different from that dominating the other. Of the difficult set, this phenomenon occurs 37.1% of the time in adjacent sentence main event links and 34.9% of the time in links where one event subordinates another. This suggests that further investigation may be fruitful. There is comparatively very little change of tense in the event-time linking tasks; none in same-sentence event-timex linking and only 5.3% for event-DCT links.

Differing modalities are very common in in task F's difficult set, as expected for cases where some events subordinate others (this is the category that *if-event-then-event* constructions typically go in), but not common at all for task E.

It is interesting to note the relative lack of shifts in dominant tense in difficult timex-event links when compared to difficult event-event links. This reflects the findings of Harris and Brewer (1973), that temporal adverbs bolster the cognitive role of verb tenses. From these observations, one could suggest that when times are known, a qualifying temporal adverb can be used in place of the information provided by a shift of tense. Validation of this hypothesis remains for future work.

Poor annotation is a potential difficulty source. TempEval-2 data is based on TimeBank, which has an IAA of only 0.77 for TLINK `relType`s. The TempEval-2 relation set is simpler than TimeBank's, so 0.77 is a minimum IAA. Investigation of the difficult set showed that the frequency

		Tense shift		
		No	Yes	Total
Signal	No	57	38	95
	Yes	10	0	10
	Total	67	38	105

Table 4.7: Co-occurrence frequencies for temporal signals and tense shifts in event-event difficult links.

		Tense shift		
		No	Yes	Total
Signal	No	1908	1303	3211
	Yes	155	115	270
	Total	2063	1418	3481

Table 4.8: Co-occurence frequencies for temporal signals and tense shifts in all TimeBank v1.2's event-event links.

of annotation disagreement was in line with what one might expect. The rate of disagreement with the relation type annotation among links in the difficult set was between 4.6% and 18.5%. This disagreement rate was consistently higher for event-time links than event-event links, but never higher than average IAA accounts for (23%), so the difficult links are probably not hard due solely to poor annotation.

4.3.3 Signals vs. tense shifts

Signals and tense shift are prevalent in the difficult set. It may be useful to investigate both these phenomena. To avoid redundant investigation, one must first establish some degree of independence between the two; if e.g. solving the relation labelling problem for links with tense shifts also solves the problem for those with signals, then it is not worth investigating both.

It has been proposed that both tense shifts and temporal adverbs provide temporal ordering cues (Harris and Brewer, 1973). Further, it is suggested that lexicalised temporal markers and tense shifts provide information independently – that is to say, there is no overlap in the information provided by either one of these. Temporal information conveyed by tense shift is independent of that provided in signals. We investigate this using empirical data and briefly test the hypothesis that they are exclusive with regard to the temporal information they provide.

Exploring further the idea of explicit temporal qualification (such as with a temporal adverbial) as an alternative to tense shifts, a brief investigation into the overlap between temporal signals and tense shifts is worthwhile. The data has been gathered and, while not excessive, 105 records (total difficult links from tasks E and F) is enough to estimate the degree of overlap. Results are shown in Table 4.7.

In the case of the difficult event-event links, there was no overlap between links where tense

shifted between arguments and links that used an explicit temporal signal. The two categories were in fact mutually exclusive. This was a significant deviation from the overlap that would occur if the two phenomena were mutually exclusive (which would be ˜6.3 TLINKs).

Looking at all event-event links in TimeBank 1.2 (difficult and non-difficult), the data is different from TempEval. The overlap between signalled and tense-shifted links is as if these phenomena are almost independent (Table 4.8). This can be demonstrated as follows. The global probability of an event-event link using a signal, $P(S)$, is 7.76%. Similarly, that of such a link using a tense shift $P(T)$ is 40.6%. If these variables are independent, $P(S \cap T) = P(S) \cdot P(T)$. We know that in the general case, $P(S \cap T) = 3.30\%$; further, $P(S) \cdot P(T) = 3.15\%$. This is close to suggesting independence.

Another test is to look for prior probabilities with Bayes' theorem. If independent of T, S with not affect $P(T)$ and vice versa. From the data, $P(T|S) = 42.6\%$ which is only 4.9% out from $P(T)$ and $P(S|T) = 8.11\%$ is even closer to $P(S)$ with a 4.5% difference.

However, for the difficult links, despite $P(S)$ and $P(T)$ having roughly similar values, $P(S \cap T) = 0$, which is significantly different from what one would expect, even after taking into account the size of the dataset. Therefore, we might say that having both a tense shift and a signal present makes a link relatively easy to automatically label. Certainly in cases where neither a tense shift not a signal appear, the relation is likely to be difficult to classify.

4.3.4 Extra-feature analysis summary

Certain properties were observed in large proportions of difficult links. Difficult event-time relations (tasks C & D) often employed a temporal signal, relied on global inference, or had differing modalities. Difficult event-event relations (tasks E & F) often relied on inference, exhibited iconicity, involved a tense or aspect shift, or had differing modalities. A large proportion of relations have explicit signal or tense/aspect annotations. As this data is directly available and affects a notable proportion of observed TLINKs, these two phenomena were selected for future investigation.

4.3.5 Next directions

This section provided a data-driven analysis of difficult TLINKs in a well-known dataset using non-surface criteria. A set of commonly-difficult links was identified for each task. Further, a set of potential temporal information sources was identified in terms of linguistic phenomena and these phenomena monitored for each difficult link. This leads to a set of candidate information types for further investigation. What remains to be done is to outline a framework for working with temporal links using these types of temporal phenomena, so that we have experimental and evaluation methods to use in investigation.

4.4 Analysing TLINKs through dataset segmentation

Our approach is to first identify the type of information used to link two entities and then to classify a relation. This section describes the core approach and then enumerates the various

special situations of links to be explored in later experimental chapters.

We are not concerned with determining which entities should be temporally linked in a discourse. We constrain our problem, as in the majority of previous work, to providing the relation type of a given entity pair.

4.4.1 Core approach

The temporal relation labelling experiments in this thesis adopt a machine-learning approach, based on that of Mani et al. (2007). Experiments are split into "situations", each of which applies to a subset of temporal links. The identification of links in a particular situation is automatic and a method given for each. Additional features are then added to the core set and a classifier learned and evaluated on the links in a situation. Performance is compared with a classifier learned over the same data but without the additional features.

The base set of features is derived directly from the TimeML attribute values, and is as follows:

- event/timex text;
- TimeML tense for each event;
- TimeML aspect for each event;
- modality for each event;
- cardinality for each event;
- polarity for each event;
- part-of-speech for each event;
- class for each event;
- document function for each timex;
- quantisation for each timex;
- frequency for each timex;
- timex value for each timex;
- temporal function for each timex;
- "mod" for each timex;
- type for each timex;
- are both relation arguments in the same sentence?;
- are both relation arguments in adjacent sentences?;
- if events, do both relation arguments have the same TimeML aspect?;
- if events, do both relation arguments have the same TimeML tense?;
- does argument 1 textually precede argument 2?

4.4.2 Theoretical assumptions

This analysis expects that expressions conveying temporal relation type are present in discourse. Also, even though each relation may be expressed in many way, we assume that it is not. If every available device above is always used to indicate a temporal relation, the analysis' results would be meaningless, as it would show that all types of information are used for all links.

Instead, the approach outlined above makes the assumption that only the minimum amount of language is used to express temporal information. That is, that information theory (Shannon, 1949) concepts such as the minimum description length (MDL) (Rissanen, 1978) will apply to languages also (as also posited by e.g. Grünwald (1996)). In this context, the MDL principle suggests that unexpected deviations from how time is described require the addition syntactic or lexical information, given a standard "temporal model" of discourse.

Examples of the principle being present in time-relation language are not difficult to come by. One may observe it in phenomena such as temporal signals, tense shifts or temporal expressions. Temporal signals are connectives that explicitly describe a certain ordering but are not required for the majority of relations (they only signal about 12% of TimeBank's links, for example). Tense shifts require a different term of expression, which may come from the insertion of auxiliary verbs or a change of inflection, and yield a new reference time, event time or even temporal relation. Each shift carries information. Finally, the length and complexity of a temporal expression can correlate to its precision or its distance from the current timeframe; "*At 8.56am on the 19th August, 2006*" is long, complex and highly specific – "*last week*" serves only to shift the timeframe for anchoring day names backwards. Changing the nominal structure of a sentence is required to express temporal phenomena again. It is this extra information, describing temporal relations, that we are attempting to identify and exploit.

4.5 Chapter Summary

This chapter used a set of empirical data to determine what constitutes a difficult temporal link, and an investigation into linguistic phenomena that occur frequently in the relations that are hardest to automatically label. For each category of relation in TempEval-2 (i.e. Tasks C-F), between 8% and 47% of temporal relations in documents were difficult for the majority of automatic systems. Event-event relations were consistently the most difficult to type: where 44%-47% of event-event links were difficult, in contrast to event-time links, for which only 8%-19% were difficult.

After an analysis of temporal relations that are difficult to label automatically, themes common in these difficult temporal relations were identified. It was found that two linguistic phenomena were particularly more prevalent in difficult relations than in the general case. First, difficult links often incorporated an explicit co-ordinating temporal signal (a word like *simultaneously* or *thereafter*). Second, shifts of tense and aspect between arguments were often present in difficult links. Other contributing factors were implicit temporal relations discoverable through inference, and changes in modality, though these were less prevalent.

Based on this analysis, the remainder of this thesis comprises two major parts: an investigation into temporal signals, and another into a framework of tense and aspect. Signals have been found to be useful. We demonstrate how they may be used for temporal relation labelling and then investigate the automatic annotation of temporal signals in Chapter 5. Models of tense can account for a whole group of situations, including reported speech, tense shifts and the use of timexes to shift the frame of reference. Such situations are detailed in Chapter 6.

Chapter 5

Using Temporal Signals

> Words are but the signs of ideas.
>
> *Preface to the Dictionary*
> Samuel Johnson

5.1 Introduction

In Chapter 4, we saw that a proportion of difficult temporal relations were associated with a particular separate word or phrase that described the temporal relation type – a **temporal signal**. The failure analysis in Section 4.3.1 finds signals to be of use in over a third of difficult TLINKs. Despite their demonstrable impact on temporal link labelling (see Section 3.5.4), no work has been undertaken toward the automatic annotation of temporal signals, and little toward their exploitation. This chapter begins to address these deficiencies.

Temporal signals (also known as temporal conjunctions) are discourse markers that connect a pair of events and times and explicitly state the nature of their temporal relation. Humans resolve events and times in discourses that machines cannot yet automatically label. It is assumed that there must be information in the document and in world knowledge that allows resolution of events, times and relations between them. Temporal signals form part of this information. Intuitively, these words contain temporal ordering information that human readers can access. This chapter investigates the role that temporal signals play in discourse and finds methods for automatically annotating them.

To illustrate:

(5.1) *"The exam papers were submitted before twelve o'clock."*

In Example 5.1 there is an event, the submitting of exam papers, and a time, twelve o'clock, that are temporally related. The word *before* serves as a signal that describes the nature of the temporal relation between them.

These temporal signals can occur with difficult temporal links and seem to provide explicit information about temporal relation type. It is worth investigating their potential utility in the

relation typing task. If these signals are found to be useful, we may determine how to detect and use them automatically, instead of relying on existing manual annotations. To begin investigation the process of automatic signal annotation, a thorough account of temporal signals is required, followed by an examination of current resources that include temporal signal annotations. Next one may cast the signal annotation problem as a two step process. Firstly, one must know how to determine which words and phrases in a given document are temporal signals. Secondly, one needs to work out with which intervals a given temporal signal is associated, given many candidates. The tasks jointly comprise automatic temporal signal annotation.

This chapter is therefore structured as follows. In Section 5.2, we formally introduce background material regarding temporal signals. Section 5.3 reports on the effect that signal information has on an existing relation typing approach compared with the approach's performance sans signal information, finding that adding features that describe temporal signals yields a large error reduction for automatic relation typing. Accordingly, after surveying signal annotations in existing corpora (Section 5.4), a method for automatically finding words and phrases that occur as temporal signals is introduced, which first requires the construction of a high-quality ground truth dataset (Section 5.5). After developing an approach to finding temporal signal expressions using this new dataset (Section 5.6), Section 5.7 describes a method for associating temporal signal (once found) with a pair of temporally-related intervals whose relation is described by the temporal signal. The overall performance of the presented temporal signal annotation system is then evaluated. The chapter concludes with an evaluation of the impact this automatic signal annotation has on the overall relation typing task (Section 5.8), which is a positive one.

5.2 The language of temporal signals

Signal expressions explicitly indicate the existence and nature of a temporal relation between two events or states or between an event or state and a time point or interval. Hence a temporal signal has two arguments, which are the temporal "entities" that are related. One of these arguments may be deictic instead of directly attached to an event or time; anaphoric temporal references are also permitted. For example, the temporal function and arguments of *after* in "*John slept after a long day at work*" are clear and are available in the immediately surrounding text. With "*After that, he swiftly finished his meal and left*" we must look back to the antecedent of *that* to locate the second argument.

Sometimes a signal will appear to be missing an argument; for example, sentence-initial signals with only one event in the sentence (*"Later, they subsided."*). These signals relate an event in their sentence with the discourse's current temporal focus – for example, the document creation time, or the previous sentence's main event.

Signal surface forms have a compound structure consisting of a **head** and an optional **qualifier**. The head describes the temporal operation of the signal phrase and the qualifier modifies or clarifies this operation. An example of an unqualified signal expression is *after*, which provides information about the nature of a temporal link, but does not say anything about the absolute or relative magnitude of the temporal separation of its arguments. We can elaborate on this magnitude with

phrases which give qualitative information about the relative size of temporal separation between events (such as *very shortly after*), or which give a specific separation between events using a duration as a modifying phrase (e.g. *two weeks after*). In both cases, the signal applies to the ordering of events either side of the separation, rather than the separation itself.

5.2.1 Related work

Signals help create well-structured discourse. Temporal signals can provide context shifts and orderings (Hitzeman, 1997). These signal expressions therefore work as discourse segmentation markers (Ho-Dac and Péry-Woodley, 2008). It has been shown that correctly including such explicit markers makes texts easier for human readers to process (Bestgen and Vonk, 1999).

Further, words and phrases that comprise signals are sometimes polysemous, occurring in temporal or non-temporal senses. For the purposes of automatic information extraction, this introduces the task of determining when a given candidate signal is used in a temporal sense.

Brée and Smit (1986) performed a study of temporal conjunctions and prepositions and suggested rules for discriminating temporal from non-temporal uses of signal expressions that fall into these classes. Their approach relies heavily upon the presentation of contrasting examples of each signal word. Brée et al. (1993) later describe the ambiguity of nine temporal prepositions in terms of their roles as temporal signals.

Schlüter (2001) identifies signal expressions used with the present perfect and compares their frequency in British and US English. This chapter later attempts a full identification of English signal expressions.

Vlach (1993) presents a semantic framework that deals with duratives when used as signal qualifiers (see above). Our work differs from the literature in that is it the first to be based on gold standard annotations of temporal semantics and that it encompasses all temporal signal expressions, not just those of a particular grammatical class.

Intuitively, signal expressions contain temporal ordering information that human readers can access easily. Once temporal conjunctions are identified, existing semantic formalisms may be readily applied to discourse semantics. It is however ambiguous which temporal relation any given signal attempts to convey, as investigated by (Hitzeman, 2005) and studied in TimeBank later in this chapter (Section 5.4.2). Our work quantifies this ambiguity for a subset of signal expressions.

5.2.2 Signals in TimeML

This section includes work from Derczynski and Gaizauskas (2011a).

TimeML's description of a signal is:[1]

> `SIGNAL` is used to annotate sections of text, typically function words, that indicate how temporal objects are to be related to each other. The material marked by `SIGNAL` constitutes the following:
>
> - indicators of temporal relations such as temporal prepositions (e.g *"on"*, *"during"*) and other temporal connectives (e.g. *"when"*) and subordinators (e.g. *"if"*).

[1] TimeML Annotation Guidelines, http://timeml.org/site/publications/specs.html.

Phrase	Corpus freq.	Occurrences as signal	Likelihood of being a signal
subsequently	3	3	100%
after	72	67	93%
's	10	8	80%
follows	4	3	75%
before	33	23	70%
until	36	25	69%
during	19	13	68%
as soon as	3	2	67%

Table 5.1: A sample of phrases most likely to be annotated as a signal when they occur in TimeBank. All corpus data was provided by the CAVaT tool (Derczynski and Gaizauskas, 2010a).

This functionality of the `SIGNAL` tag was introduced by Setzer and Gaizauskas (2000).

- indicators of temporal quantification such as *"twice"*, *"three times"*.

Signals in TimeML are used to mark words that indicate the type of relation between two intervals and also to indicate multiple occurrences of events (temporal quantification). For the task of temporal relation typing, we are only interested in this former use of signals. The annotation guidelines suggest that in TimeML one should annotate a minimal set of tokens – typically just the "head" of the signal.

For example, in the sentence *John smiled after he ate*, the word *after* specifies an event ordering. Example 5.2 shows this sentence represented in TimeML.

(5.2)
```
John <EVENT id="e1"> smiled </EVENT> <SIGNAL id="s1"> after </SIGNAL>
he <EVENT id="e2"> ate </EVENT> .
<TLINK id="l1" eventID="e1" relatedToEvent="e2"
  relType="AFTER" signalID="s1" />
```

TimeML allows us to associate text that suggests an event ordering (a `SIGNAL`) with a particular temporal relation (a `TLINK`). To avoid confusion, it is worthwhile clarifying our use of the term "signal". We use **SIGNAL** in capitals for tags of this name in TimeML and **signal/signal word/signal phrase** for a word or words in discourse that describe the temporal ordering of an event pair. Examples of the signals found in TimeBank are provided in Table 5.1.

It is important to note that not every occurrence of text that could be a signal is used as a temporal signal. Some signal words and phrases are polysemous, having both temporal and non-temporal senses: e.g. *"before"* can indicate a temporal ordering (*"before 7 o'clock"*) or a spatial arrangement (*"kneel before the king"*). This thesis refers to expressions that could potentially be temporal signals as **candidate signal phrases**. Only candidate signal phrases occurring in a temporal sense are of interest.

Signal Expression	TLINK count	AFTER	BEFORE	BEGINS	BEGUN_BY	DURING	ENDED_BY	ENDS	IAFTER	IBEFORE	INCLUDES	IS_INCLUDED	SIMULTANEOUS
after	76	62	3	4				5	2				
when	57	16	3	1	2	1			1	1	9	9	14
until	37	4	7	1			21	1	1	2			
before	36	1	28	2			1	2	1	1			
since	19	9	1	2	7								
already	13		6								4	3	
previously	18	6	12										
while	9												9
meanwhile	9		1			2						1	5
followed	4	2	2										
former	12		12										

Table 5.2: Signal expressions and the TimeML relations that they can denote. Counts do not match because a single signal expression can support more than one temporal link.

The signal text alone does not mean a single temporal interpretation. A temporal signal word such as *after* (for example) is used in TimeBank in `TLINK`s labelled AFTER, BEFORE and INCLUDES. For example, there is no set convention to the order in which a `TLINK`'s arguments should be defined; the AFTER `TLINK` in Example 5.2 could just as well be encoded as:

```
<TLINK id="l1" eventID="e2" relatedToEvent="e1"
  relType="BEFORE" signalID="s1" />
```

See Table 5.2 for the distribution of relation labels described by a subset of signal words and phrases.

As described above, signals sometimes reference abstract points as their arguments. These abstract points might be a reference time (Section 6.3) or an implicit anaphoric reference. As TimeML does not include specific annotation for reference time, one should instead assume that the signal co-ordinates its non-abstract argument with the interval at which reference time was last set. For example, in *"There was an explosion Tuesday. Afterwards, the ship sank"*, we will link the *sank* event with *explosion* (the previous head event) and then associate our signal with this link.

5.3 The utility of temporal signals

Do signals help temporal relation typing? Given the role that they might play in the relation typing task suggested in Section 4.3.1 and having a high-level definition of temporal signals, it is next important to establish their potential utility. Since we have in TimeML a signal-annotated corpus, to answer this question, one can compare the performance of automatic relation typing

Corpus	Total TLINKs	With SIGNAL		Without SIGNAL
TimeBank v1.2	6418	718	(11.2%)	5700
AQUAINT TimeML v1.0	5365	178	(3.3%)	5187
ATC (combined)	11783	896	(7.6%)	10887
ATC event-event	6234	319	(5.1%)	5915

Table 5.3: TLINKs and signals in the largest TimeML-annotated corpora.

systems with and without signal information. Positive results would motivate investigation into further work on automatic signal annotation. This section relates such a comparison, and includes work from (Derczynski and Gaizauskas, 2010c). An extended investigation into this section's findings can be found in (Derczynski and Gaizauskas, 2013b).

Although accurate event ordering has been the topic of research over the past decade, most work using the temporal signals present in text has been only preliminary. However, as noted in Chapter 3, specifically focusing on temporal signals when classifying temporal relations can yield a performance boost. This section attempts to measure that performance boost.

In TimeML, a signal is either text that indicates the cardinality of a recurring event, or text that explicitly states the nature of a temporal relation. Only the latter sense is interesting for the current work. This class of words and phrases includes temporal conjunctions (e.g. *after*) and temporal adverbials (e.g. *currently*, *subsequently*), as well as set phrases (e.g. *as soon as*). A minority of `TLINK`s in TimeML corpora are annotated with an associated signal (see Table 5.3).

While the processing of temporal signals for `TLINK` classification could potentially be included as part of feature extraction for the relation typing task, temporal signals are complex and useful enough to warrant independent investigation. When the final goal is `TLINK` labelling, once salient features for signal inclusion and representation have been found, one might skip signal annotation entirely and include these features in a temporal relation type classifier. As we are concerned with the characterisation and annotation of signals, we do not address this possibility here, instead attempting to understand signals as an intermediate step towards better overall temporal labelling.

The following experiment explores the question of whether signal information can be successfully exploited for `TLINK` classification by contrasting relation typing with and without signal information. The approach replicated as closely as possible is that of Mani et al. (2006), briefly summarised as follows.

The replication had three steps. Firstly, to simplify the problem, the set of possible relation types was reduced (folded) by applying a mapping (see Section 3.3.1). For example, as `a` BEFORE `b` and `b` AFTER `a` describe the same ordering between events `a` and `b`, we can flip the argument order in any AFTER relation to convert it to a BEFORE relation. This simplifies training data and provides more examples per temporal relation class. Secondly, the following information from each `TLINK` is used as features: event class, aspect, modality, tense, negation, event string for each event, as well as two boolean features indicating whether both events have the same tense or same aspect.

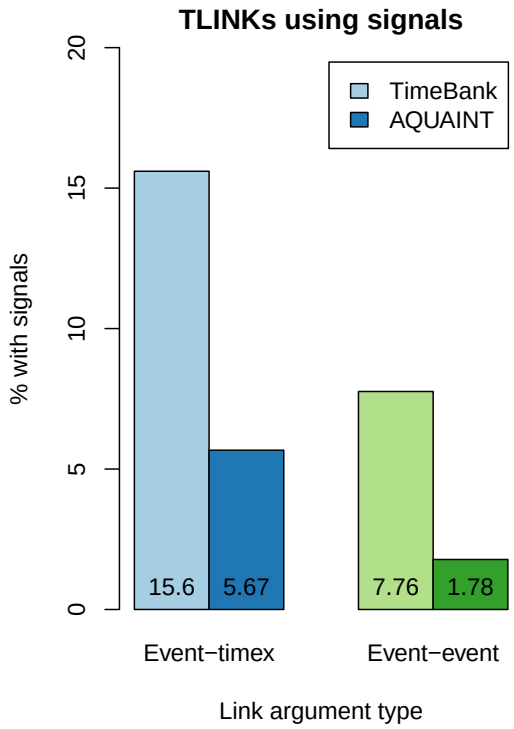

Figure 5.1: Signalled TLINKs by argument type (event-event or event-tlink) in TimeBank 1.2 and the AQUAINT TimeML corpus. The paler columns correspond to TimeBank, the darker AQUAINT.

	Corpus	XV accuracy	Train/Eval split	Baseline
Mani et al. results	AQ + TimeBank 1.2a	61.79%		51.6%
Replicated results	AQ + TimeBank 1.2	60.32%	60.04%	53.34%

Table 5.4: Results from replicating a prior experiment on automatic relation typing of event-event relations.

Thirdly, we trained and evaluated the predictive accuracy of the maximum entropy classifier from Carafe.[2] To match the original approach, ten-fold cross-validation was used, and a one-third/two-thirds split was also introduced to see the effect of reduced ratio of training:evaluation examples. This split the set of event-event `TLINK`s into a training set of 4156 instances and an evaluation set of 2078 instances.

In Mani et al. (2006), TLINK data came from the union of TimeBank v1.2a and the AQUAINT TimeML corpora. As the TimeBank v1.2a corpus used is not publicly available, we used TimeBank v1.2. This use of a publicly-available version of TimeBank instead of a private custom version was the only change from the previous work. In this work we only examine event-event links, which make up 52.9% of all TLINKs in our corpus, likely due to minor differences between the TLINK annotations of TimeBank v1.2 and TimeBank v1.2a.

Table 5.4 shows results from replicating the previous experiment on event-event TLINKs. The

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

baseline listed is the most-common-class in the training data. This gives a similar score of 60.32% accuracy compared to 61.79% in the previous work. The differences may be attributed to the non-standard corpus that they use. The TLINK distribution over a merger of TimeBank v1.2 and the AQUAINT corpus differs from that listed in the paper.

5.3.1 Introducing signals to the relation labelling feature set

Now that a reasonable replication of a prior approach has been established, the goal is to measure the difference in relation typing performance that temporal signals make. This requires feature representations of signals. To add information about signals to our training instances, we use the extra features described below; the two arguments of a TLINK are represented by **e1** and **e2**. All features can be readily extracted from the existing TimeML annotations. Only gold-standard signal annotations from the corpora were used.

- **Signal phrase.** This shows the actual text that was marked up as a SIGNAL. From this, we can start to guess temporal orderings based on signal phrases. However, just using the phrase is insufficient. For example, the two sentences *Run before sleeping* and *Before sleeping, run* are temporally equivalent, in that they both specify two events in the order run-sleep, signalled by the same word *before*.
- **Textual order of e1/e2.** It is important to know the textual order of events and their signals even when we know a temporal ordering. Textual order can have a direct effect on the temporal order conveyed by a signal. To illustrate, *"Bob washes before he eats"* describes a story different from *"Before Bob washes he eats"*.
- **Textual order of signal and e1, signal and e2.** These features describe the textual ordering of both TLINK arguments and a related signal. It will also help us see how the arguments of TLINKs that employ a particular signal tend to be textually distributed. The features are required to disambiguate cases where textual order is unreliable. To illustrate, *"Bob washes before he eats"* and *"Before he eats, Bob washes"* describe the same event ordering but have different text orderings.
- **Textual distance between e1/e2.** Sentence and token count between e1 and e2.
- **Textual distance from e1/e2 to SIGNAL.** If we allow a signal to influence the classification of a TLINK, we need to be certain of its association with the link's events. Distances are measured in tokens.
- **TLINK class given SIGNAL phrase.** Most likely TLINK classification in the training data given this signal phrase (or empty if the phrase has not been seen). Referred to as signal **hint**. Referred to as signal **hint**.

5.3.2 TLINK typing results using signals

Table 5.5 shows the results of adding features for temporal signals to the basic TLINK relation typing system. Moving to a feature set which adds SIGNAL information, including signal-event word order/distance data, 61.46% predictive accuracy is reached. The increase is small when

Predictive accuracy	XV	Split
Baseline (most common class)	53.34%	53.34%
Without signal features	60.32%	60.04%
With basic signal features	**61.46%**	60.81%
With signal features including hint	n/a	**61.98%**

Table 5.5: TLINK classification with and without signal features, using both 10-fold cross validation and a one-third/two-thirds split between evaluation and training data.

	Cross validation		**Train/Eval split**	
Predictive accuracy	**Unsignalled**	**Signalled**	**Unsignalled**	**Signalled**
Baseline (most common class)	52.68%	62.41%	52.68%	62.41%
Plain features	**62.05%**	55.65%	61.81%	60.32%
Plain, signal features	62.05%	**69.57%**	61.81%	**82.19%**
Plain, signal features, hint	62.05%	41.72%	-	-

Table 5.6: Predictive accuracy from Carafe's maximum entropy classifier, using features that do or do not include signal information, over signalled and non-signalled event-event TLINKs in ATC. The baseline is accuracy when the most-common-class is always assigned.

compared to 60.32% accuracy without this information, but TLINKs that employ a SIGNAL in are a minority in our corpus (possibly due to under-annotation).

The low magnitude of the performance increase seen in Table 5.5 could be due to the way in which training examples are selected. There are in total 11 783 TLINKs in the combined corpus, of which 7.6% are annotated including a SIGNAL; for just TimeBank v1.2, the figure is higher at 11.2% (see Table 5.3 and also Figure 5.1). The proportion of signalled TLINKs in our data – event-event links in the combined AQUAINT/TimeBank 1.2 corpus – is lowest at 5.1%. It is possible that signalled TLINKs are classified significantly better using this extended feature set, but account for such a small part of this dataset that the overall difference is small. To test this, the experiment is repeated, this time splitting the dataset into signalled and non-signalled TLINKs.

If there is no performance difference between feature sets when classifying TLINKs that *do* use signals, then our hypothesis is incorrect, or the features used are insufficiently representative. If signals are helpful, and our features capture information useful for temporal ordering, we expect a performance increase when automatically classifying signalled TLINKs. Results in Table 5.6 support our hypothesis that signals are useful, but we are performing nowhere near the maximum level suggested above. Data sparsity is a problem here, as the combined corpus only contains 319 suitable TLINKs, and both source corpora show evidence of signal under-annotation. The results also suggest that the signal hint feature was not helpful; this is the same result found by Bethard et al. (2007b).

Exploring the strongest feature set (basic+signals; no hint), and attempting to combat the data sparsity problem, we used 10-fold cross validation instead of a split; results are also in Table 5.6.

This again shows a distinct improvement in the predictive accuracy of signalled TLINKs using this feature set over the features in previous work. Cross-validation also gives better overall accuracy. This is likely because of the low volumes of training data mean that the real difference in number of examples between 10-fold cross validation and a one-third/two-thirds split can make a large contribution to classifier performance.

5.3.3 Utility Assessment Summary

When learning to classify signalled TLINKs, there is a significant increase in predictive accuracy when features describing signals are introduced. This suggests that signals are useful when it comes to providing information for classifying temporal links, and also that the features we have used to describe them are effective.

Now that it is confirmed that signals are helpful in temporal relation typing, the next task is to determine how to annotate them automatically. A good account of existing resources may give clues for this process. After this, one needs to explore how to discriminate whether or not a candidate signal expression is used as a temporal signal in text. Next, after finding a temporal signal, we need to determine which intervals it temporally connects. Finally, we can attempt to annotate a temporal link based on the signal.

5.4 Corpus Analysis

In order to understand temporal signals, this section investigates the role of hand-annotated temporal signals in the TimeBank dataset. Further, casual examination reveals that words acting in a temporal signal role in existing datasets are not always annotated as such. Under-annotation can depend on how well the annotator understands the task, and the clarity of annotation guidelines. This section discusses the TimeML definition of signals and describes an augmented corpus which has received extra annotation.

Using the TimeBank corpus, we set out to answer the following questions:

1. Of the expressions which can function as temporal signals, what proportion of their usage in the TimeBank corpus is as a temporal signal? E.g. how ambiguous are these expressions in terms of their role as temporal signals?

2. Of the occurrences of these expressions as temporal signals, how ambiguous are they with respect to the temporal relation they convey?

The following section (which includes material from Derczynski and Gaizauskas (2011a)) provides provisional answers to these questions – provisional as one of the difficulties we encountered was significant under-annotation of temporal signals in TimeBank. We have addressed this to some extent, but more work remains to be done. Nonetheless we believe the current study provides important insights into the behaviour of temporal signals and how they may be exploited by computational systems carrying out the temporal relation detection task.

Annotated SIGNAL elements	758
Signals used by a TLINK	721
Signals used by an ALINK	1
Signals used by a SLINK	39
TLINKs that use a SIGNAL	787
Signals used by more than one TLINK	54

Table 5.7: How <SIGNAL> elements are used in TimeBank.

Argument pairs co-ordinated	Frequency
1	597
2	41
3	12
5	1

Table 5.8: The number of TLINKs associated with each temporal signal word/phrase, in TimeBank. Signals not used on TLINKs (e.g. those used on aspectual or subordinate links, or for event cardinality) are excluded. The distribution appears to be Zipfian (Zipf, 1935).

5.4.1 Signals in TimeBank

The TimeML <SIGNAL> element bounds a lexicalised temporal signal. Summary information on the SIGNAL elements in TimeBank 1.2 is in Table 5.7 and the number of links per signal in Table 5.8. Although permitted under TimeML 1.2.1 for denoting cardinality, no signals have been assigned to event instances for this purpose, although there is one unassigned signal annotation that does indicate event cardinality.

In cases where a specific duration occurs as part of a complex qualifier-head temporal signal, e.g. *two weeks after*, TimeBank has followed the convention that the signal head alone is annotated as a SIGNAL and the qualifier is annotated as a TIMEX3 of type DURATION.

5.4.2 Relation Type Ambiguity

The nature of the temporal relation described by a signal is not constant for the same signal phrase, though each signal tends to describe a particular relation type more often than other types. Table 5.2 gives an excerpt of data showing which temporal relations are made explicit by each signal expression. The variation in relation type associated with a signal is not as great as it might appear as the assignment of temporal relation type has an element of arbitrariness – one may choose to annotate a BEFORE or AFTER relation for the same event pair by simply reversing the temporal link's argument order, for example. There is no TimeML convention regarding how TLINK annotation arguments should be ordered. Nevertheless, it is possible to draw useful information from the table; for example, one can see that *meanwhile* is much more likely to suggest some sort of temporal overlap between events than an ordering where arguments occur discretely.

after	ensuing	meantime	soon
afterwards	eventually	momentarily	still
again	fifthly	next	subsequent
already	finally	ninethly	subsequently
as	first	now	succeeding
as soon as	firstly	nowadays	suddenly
as yet	following	on	supervening
at	for	once	then
at once	forever	originally	thereafter
at this point	for ever	over	thirdly
before	former	past	through
beforehand	formerly	preceding	throughout
between	fourthly	presently	til
by	frequently	previous	till
coexisting	from	previously	to
coinciding	here	prior	up to
concurrent	hitherto	recently	until
concurrently	immediately	secondly	when
contemporaneous	in	seventhly	whenever
contemporaneously	initially	shortly	while
contemporary	instantly	simultaneous	whilst
directly	last	simultaneously	within
during	late	since	yet
earlier	lately	sixthly	's
early	later	so long as	
eighthly	meanwhile	sometime	

Table 5.9: A closed class of temporal signal expressions

Part of speech	Frequency	Proportion
IN	521	77.3%
RB	73	10.8%
WRB	53	7.9%
JJ	14	2.1%
RBR	5	0.7%
VBG	4	0.6%
CC	2	0.3%
RP	1	0.1%
JJR	1	0.1%

Table 5.10: Distribution of part-of-speech in signals and the first word of signal phrases.

Closed class of signals

To what extent are the words sometimes annotated as temporal signals in TimeBank actually used as time relaters?

As temporal signals and phrases are likely to be a closed class of words, our approach is to first define a set of temporal signal candidate words. For each occurrence of one of these words in a discourse, we will decide if it is a temporal signal or not.

Because they do not contribute to temporal ordering, annotated signals that indicate the cardinality of recurring events were removed before experimentation. We have derived a closed class of 102 signal words and phrases from Quirk et al. (1985) (see for example Section 10.5, "Time Relaters"), given in Table 5.9. This list is long but may not be comprehensive. Automatic signal annotation can be approached by finding words in a given document that are both within this closed class of candidate signal phrases and also occur having a temporal sense. TimeBank contains 62 unique signal words and phrases (ignoring case), annotated in 688 `SIGNAL` elements and used by 718 `TLINK`s. Of these 62, over half (39) are also found in our list above. The remaining 23 signals correspond to only 45 signal mentions, supporting 46 temporal links. Thus, if we can perfectly annotate every signal we find in text based on our closed class, we will have described 93.1% of `TLINK`-supporting signals and be better able to label 93.6% of `TLINK`s that have a supporting signal.

To provide a surface characterisation of the role signals play, the distribution of their part of speech tag (from PTB) over signals in TimeBank is given in Table 5.10. Many uses are as prepositions, perhaps for attaching events to each other by means of prepositional phrases.

Of the closed class entries detailed in Table 5.9, 25 entries occur in the corpus but are never annotated as signal text: *again*, *directly*, *early*, *finally*, *first*, *here*, *last*, *late*, *next*, *now*, *recently*, *eventually*, *forever*, *formerly*, *frequently*, *initially*, *instantly*, *meantime*, *originally*, *prior*, *shortly*, *sometime*, *subsequent*, *subsequently* and *suddenly*.

We could also derive an alternative signal list by extracting all phrases that are found as the first child of `SBAR-TMP` constituent tags, as suggested in Dorr and Gaasterlaand (Dorr and Gaasterland, 2006). For example, in Example 5.2 (an automatically parsed and function-tagged

(5.3)

```
(S1 (S (NP-SBJ (NNP Nashua))
       (VP (VBD announced)
         (NP (DT the) (NNP Reiss) (NN request))
         (SBAR-TMP (IN after)
           (S (NP-SBJ (DT the) (NN market))
             (VP (VBD closed))))) (. .)))
```

Figure 5.2: An example SBAR-TMP construction around a temporal signal

Correct examples	Incorrect examples
after	at least
as	as surely
before	several months
once	nearly two months
since	even
until	only
while	soon
when	

Table 5.11: The set of signal words and phrases suggested by the SBAR-TMP model, broken into correctly and incorrectly detected phrases.

sentence from TimeBank's `wsj_0520.tml`), the first child of the `SBAR-TMP` constituent is a one-leaf IN tag. The text is *after*, which we would treat as a temporal signal. This approach returns a restrictive set of temporal signals, shown in Table 5.11, though contains few false positives.

5.4.3 Temporal vs non-temporal uses

The semantic function that a temporal signal expression performs is that of relating two temporal entities. However, the words that can function as temporal signals also play other roles.

For example, one may use *before* to indicate that one event happened temporally prior to another. This word does not always have this meaning.

(5.4) *"I will drag you before the court!"*

In Example 5.4, the reading is that one will be summoned to appear in front of the court – the spatial sense – and not that the reader will be dragged, and then later the court will be dragged. It is important to know the correct sense of these connective words and phrases.

Of all temporal relations (`TLINK`s) in the English TimeBank, 11.2% use a temporal signal in the original annotation (Table 5.3). It is important to note that some instances of signal expressions are used by more than one temporal link; see Table 5.8 for details. The most frequent signal word was "in", accounting for 24.8% of all signal-using `TLINK`s. However, only 13.3% of occurrences of

the word "in" have a temporal sense. The word "after" is far more likely to occur in a temporal sense (91.7% of all occurrences).

As an aside, the notion that temporal signals might be easily picked out based upon word class may be dispelled by examining the distribution of parts-of-speech possessed by temporal signals – see Table 5.10. Part of speech is not a reliable disambiguator of sense, in this case.

5.4.4 Parallels to Spatial Representations in Natural Language

Time and space are related and often an event will be positioned in both. Language used for describing time and language used for describing space are often similar, not least in the fact they they both use signals and often even use the same words as signals. Temporal signals relate a pair of temporal intervals, and spatial signals relate a pair of regions. Although not the focus of this chapter, it is useful to note the common and contrasting behaviours of temporal and spatial signals that emerged during investigation.

SpatialML (Mani et al., 2008) is an annotation scheme for spatial entities and relations in discourse.[3] Among other things it includes elements for annotating relations between spatial entities.

Links in SpatialML may be topological or relative. Topological links include containment, connection and other links from a fixed set based on the RCC8 calculus. SpatialML relative links, on the other hand, express spatial trajectories between locations.

In the revised ACE 2005 SpatialML annotations,[4] 97.5% of all `RLINK`s (the SpatialML representation for a relative spatial link) have at least one accompanying textual signal (See Table 5.12). Compared to TimeBank's 11.2% of TLINKs having a signal, SpatialML relative links are much more likely to use an explicit signal than TimeML temporal relations. This may be because the mechanisms available in language for expressing temporal relations are wider than those for relating spatial entities. For example, to relate events in English, one may choose to use a tense and aspect (which involves inflection or added auxiliaries) instead of adding a signal word. Furthermore, there are three spatial dimensions in which to describe an entity; in contrast, the arrow of time supplied a single unidirectional dimension, which limits range of movements and relations available.

Unlike with relative links, signal usage is lower with topological links. Only 1.85% of the latter use a signal. This distinction between relative and the temporal equivalent of topological links is not made in TimeML.

This difference in signal usage rate between topological and relative links may be because topological links are used to express relations that we infer from world knowledge and do not lexicalise. In *"A Ugandan village"*, one does not need to explain that the village is in Uganda. Relative links define one region relative to another. The nature of the relation is not easy to discern and so needs to be made explicit.

Because of the dominance of spatio-temporal sense frequencies over other uses of many of the

[3] Although SpatialML has now been superseded by ISO-Space, we are concerned in this section with a SpatialML annotated corpus; there is no ISO-Space equivalent at the time of writing.

[4] LDC catalogue number LDC2011T02

Link type	SpatialML element	Occurrences	Signalled	Signalling rate
Relative	RLINK	80	78	97.5%
Topological	LINK	378	7	1.85%

Table 5.12: Frequency of signal usage for different types of spatial link in the ACE 2005 English SpatialML Annotations Version 2.

words in this class, work on temporal signals may provide insights for future researchers working on determining spatial labels using spatial signals. This chapter will later (Section 5.6.4) on show how indications of spatial signal usage help discern temporal from non-temporal candidate signal words.

5.5 Adding Missing Signal Annotations

Given an idea of what signals are and evidence of their utility in temporal relation typing, the next step was to attempt automatic signal annotation. This was a two stage process, first concerned with identifying signal expressions that occur in a temporal sense, and then with determining which pair of events/timexes any given temporal signal co-ordinates. A preliminary approach to finding temporal signal expressions found that the dataset used suffered from low annotation quality, and so after outlining the preliminary approach, this section focuses on how the resources could be (and were) improved.

Upon examination of the non-annotated instances of words that usually occur as a temporal signal (such as *after*) it became evident that TimeBank's signals are under-annotated. In an effort to boost performance, and as there is evidence of annotation errors in the source data, we revisited the original annotations.

This chapter outlines the signal expression discrimination task only briefly, instead focusing on corpus re-annotation. The next section is dedicated entirely to the discrimination problem.

5.5.1 Preliminary signal discrimination

The overall problem is to find expressions in documents that occur as temporal signals (a fuller problem definition is given below, in Section 5.6). This was approached by considering all occurrences of expressions from the above closed class of expressions (e.g. candidate signals) and judging, for each instance, whether or not it had a temporal sense. Judgement was performed by a supervised classifier (maximum entropy), trained and evaluated using cross-validation, based on the features listed in Section 5.6.4.

Failure analysis of this initial approach suggested that the corpus was too poorly annotated to serve either as representative, solid training data for signal discrimination, or for an evaluation set for a signal discrimination approach. Some re-annotation was necessary to improve the quality of the ground truth data. This section relates the approach to, and results of, that re-annotation.

5.5.2 Clarifying signal annotation guidelines

Given that the signal annotations in TimeBank are not of sufficient quality, there are three potential causes for this: annotator fatigue, insufficient annotation guidelines, or a poor definition of signals. As annotator fatigue depends on the method of an individual annotation exercise, and TimeML's signal definition is sufficient, we seek to clarify the annotation guidelines.

To clarify the guidelines, it's important to have a thorough definition of temporal signals. While TimeML's definition is sufficient, this chapter offers an extended definition of temporal signals in Section 5.2 above.

Signal surface forms have a compound structure of a **head** and an optional **qualifier**. The head describes the general action of the signal phrase and may optionally have an attached modifying phrase. Only the head should be annotated.

(5.5) *"I arrived long after the party had finished."*

In Example 5.5, the word *after* is annotated, and the qualifier *long* is not. This would be annotated in TimeML something like:

`I arrived long <SIGNAL>after</SIGNAL> the party had finished.`

Further, a temporal signal has two arguments, which are timexes or events which are temporally related. Often both of these are explicit in the text immediately surrounding the signal. However, one may be elsewhere, as an implied argument.

5.5.3 Curation procedure

The goal is to create a firm ground truth for further investigation. Given the extended definition of a signal and the guideline clarifications just mentioned, this section details the ensuing exercise of hand-curating TimeBank to repair signal annotations.

A subset of signal words was selected for re-annotation. All instances of these words (both as temporal and non-temporal) were re-annotated with TimeML, adding `EVENT`s, `TIMEX3`s and `SIGNAL`s where necessary to create a signalled `TLINK`. We will reference this version of TimeBank with curated signal annotations as **TB-sig**.

Evaluating correct classifications against erroneous reference data will lead to artificially decreased performance. To verify that the training data (which is also evaluation data for cross-validation) is from a correct annotation, negative examples of signal words were checked manually. False negatives are removed by annotating them as TimeML signals, associating them with the appropriate TLINK or adding TLINKs and EVENTs where necessary.

Checking the entire corpus would be an exhaustive exercise. To increase the chance of finding missing annotations while limiting the search space during annotation, potentially high-impact signal words were prioritised. These were drawn from a set of signal phrases that fit the following criteria: (a) more than 10 instances in the corpus, and at least one of: (b) accuracy on positive examples less than 50% or (c) accuracy on negative examples less than 50% or (d) below-baseline classification performance. The data from this second pass is in Table 5.13.

Signal	Count	As sig.	Acc.	Change	tp	fn	fp	tn	+ve acc.
for	621	8.2%	92.4%	8%	18	33	14	556	35.3%
by	356	5.6%	95.2%	15%	7	13	4	332	35.0%
while	39	23.1%	79.5%	11%	1	8	0	30	11.1%
from	366	5.2%	94.8%	0%	2	17	2	345	10.5%
when	62	85.5%	85.5%	0%	53	0	9	0	100.0%
still	35	11.4%	88.6%	0%	0	4	0	31	.0%
already	32	40.6%	56.2%	-8%	1	12	2	17	7.7%
at	311	4.8%	94.9%	-7%	2	13	3	293	13.3%
as	271	6.6%	93.0%	-6%	3	15	4	249	16.7%
over	59	22.0%	71.2%	-31%	7	6	11	35	53.8%
since	31	58.1%	48.4%	-23%	12	6	10	3	66.7%
then	23	21.7%	73.9%	-20%	0	5	1	17	.0%
earlier	50	12.0%	86.0%	-17%	0	6	1	43	.0%
before	33	93.9%	87.9%	-100%	29	2	2	0	93.5%
previously	19	84.2%	68.4%	-100%	13	3	3	0	81.2%
former	16	75.0%	50.0%	-100%	5	7	1	3	41.7%

Table 5.13: Signal texts that are hard to discriminate; error reduction performance compared to the most common class ("change") is based on a maximum entropy classifier, trained on TimeBank. tp/fn/fp/tn correspond to counts of true and false positives and negatives.

5.5.4 Signal re-annotation observations

During curation, some observations were made regarding specific signal expressions. In some cases, these observations led to the suggestion of a feature that may help discriminate temporal and non-temporal uses of a certain expression. This section reports those observations.

Previously TimeBank contains eight instances of the word *previously* that were not annotated as a signal. Of these, all were being used as temporal signals. The word only takes one event or time as its direct argument, which is placed temporally before an event or time that is in focus. For example:

"X reported a third-quarter loss, citing a previously announced capital restructuring program"

In this sentence, the second argument of *previously* is *"announced"*, which is temporally situated before its first argument (*"reported"*). When *previously* occurs at the top of a paragraph, the temporal element that has focus is either document creation time or, if one has been specified in previous discourse, the time currently in focus.

After Of the nineteen instances of this word not annotated as temporal, only three were actually non-temporal. The cases that were non-temporal were a different sense of the word. The temporal signals are adverbial, with a temporal function. Two non-temporal cases used a positional sense. The last case was in a phrasal verb *to go after*; *"whether we would go after attorney's fees"*.

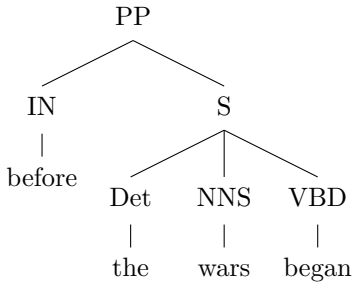

Figure 5.3: An example of the common syntactic surroundings of a *before* signal.

Throughout All the cases of *throughout* not marked as signals were not temporal signals. Four were found in the newswire header, which carries meta-information in a controlled language heavily laden with acronyms and jargon and is not prose.

Early Three of the negative instances of *early* are possibly not correctly annotated; the other 32 negatives are accurate. Of these three, one has a signal use, in part of a longer signal phrase *"as early as"*. The remaining two cases look like temporal signals. However, they are adjectival and only take one argument; there is no comparison, so we cannot say that the argument event is earlier than anything else. For this reason, they are deemed correctly annotated as non-signals.

When There are 35 annotated and 27 non-annotated occurrences of this phrase. It indicates either an overlap between intervals, or a point relation that matches an interval's start. Twenty-three of the twenty-seven non-annotated occurrences are used as temporal signals. Two of the remaining four are in negated phrases and not used to link an interval pair. for example, *"did not say <u>when</u> the reported attempt occurred"*. The other two are used in context setting phrases, e.g. *"we think he is someone who is capable of rational judgements <u>when</u> it comes to power"* (where *when it comes to* occurs in the sense of *"with regard to"*), which are not temporal in nature.

While The cases of *while* that have not been annotated as a signal – the majority class, 33 to 6 – are often used in a contrastive sense. This does suggest that the connected events have some overlap, often between statives. For example, *"But <u>while</u> the two Slavic neighbours see themselves as natural partners, their relations since the breakup of the Soviet Union have been bedeviled"*. As two states described in the same sentences are likely to temporally overlap and any events or times outside or bounding these states will be related to the state, it is unlikely that any contribution to `TLINK` annotation would be made by linking the two states with a "roughly simultaneous" relation; the closest suitable label is TempEval's OVERLAP relation Verhagen et al. (2010).

(5.6) *"nor can the government easily back down on promised protection for a privatized company <u>while</u> it proceeds with . . . "*

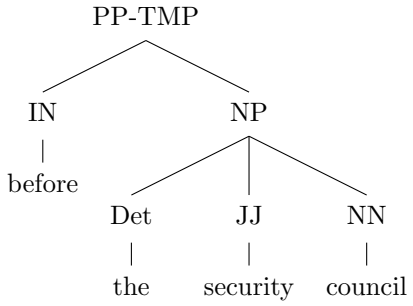

Figure 5.4: Typical mis-interpretation of a spatial (e.g. non-temporal) usage of *before*. The whole sentence was: *"The procedures are due to go before the Security Council next week."*

The cases of *while* that were not of this sense were easier to annotate. Sometimes it was used as a temporal expression; *"for a while"*. Other times, it was not used in a contrastive sense, but instead modal – see Example 5.6. The four cases of non-contrastive usage were annotated as temporal signals.

Before Three of the ten negative examples are correctly annotated. They are *before* in the spatial sense of "in front of" (as in *"The procedures are to go before the Security Council next week"*) and also a logical before that does not link instantiated or specific events (*"before taxes"*). The remaining seven unannotated examples of the word are all temporal signals. These directly precede either an NP describing a nominalised event, or directly precede a subordinate clause (e.g. `(IN before, S)` – see Figure 5.3).

Both cases of *before* that were not temporal signals were parsed and function tagged as if they were.[5]They were given the structure `(PP-TMP, (IN before) ... )` as shown in Figure 5.4.

Until All fourteen non-annotated instances of *until* should have been annotated as temporal signals. This word suggests a TimeML IBEFORE relation, unless qualified otherwise by something like "not until" or "at least until".

Already There were thirteen positive examples of *already*. All of the non-annotated examples had a non-temporal sense as per our description of temporal signals. The word tends to be used for emphasis, but can also suggest a broad "BEFORE DCT" position, which goes without saying for any past and present tensed events. As *already* can be removed without changing the temporal links present in a sentence, no further examples of this were annotated beyond the thirteen present in TimeBank.

Meanwhile This word tends to refer to a reference or event time introduced earlier in discourse, often from the same sentence. As well as a temporal sense, it can have a contrastive "despite"-like meaning. It is often used to link state-class events, which are difficult to link unless one of their

[5]Using the PTB trained Stanford Parser and the Blaheta function tagger; see Section 5.6.3

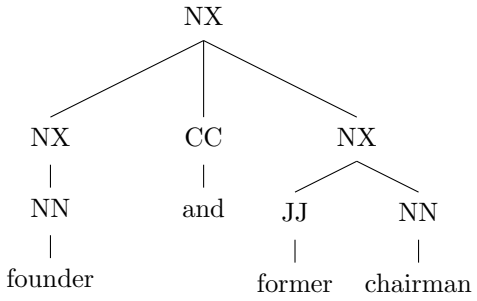

Figure 5.5: Example of a non-annotated signal (*former*) from TimeBank's `wsj_0778.tml`.

bounds is specific (see Example 5.7). In this case, it is not possible to describe the nature of the relation between the start and endpoints of either event interval, and so *meanwhile* suggests some kind of temporal overlap but nothing more. Sometimes *meanwhile* is used with no previous temporal reference. In these cases, the implicit argument is DCT. Five of the ten non-annotated *meanwhile*s were temporal signals.

(5.7) Obama was president. Meanwhile, I was a musician.

Again This word shows recurrence and is always used for this purpose where it occurs in TimeBank not annotated as a temporal signal. No instances of *"again"* were annotated.

Former This word indicates a state that persisted before DCT or current speech time and has now finished. Generally the construction that is found is an NP, which contains an optional determiner, followed by *former* and then a substituent NP which may be annotated as an `EVENT` of class STATE. This configuration suggests a `TLINK` that places the event BEFORE the state's utterance.

(5.8) *"The San Francisco sewage plant was named in honour of former President Bush."*

In Example 5.8, there is a STATE-class event – *President* – that at one time has applied to the named entity *Bush*. The signal expression *former* indicates that this state terminated BEFORE the time of the sentence's utterance.

Three-quarters of the non-annotated instances of *former* in TimeBank are temporal signals. An example non-temporal occurrence is shown in Figure 5.5

Recently Although *recently* is a temporal adverb, it cannot be applied to posterior-tensed verbs (using Reichenbach's tense nomenclature (Reichenbach, 1947)). In the corpus, these are only seen in reported speech or of verbal events that happened before DCT. *Recently* adds a qualitative distance between event and utterance time, but is of reduced use when we can already use tense information.

The phrase "Until recently" appears awkward when cast as a temporal signal but can be interpreted as "BEFORE DCT", with the interval's endpoint being close to DCT. In this case, recently functions as a temporal expression, not a signal.

Only one of the non-annotated *recently*s in TimeBank is a temporal signal. The exception, *"More recently"*, includes a comparative and is annotated as a `TIMEX3`; both this phrase and, e.g., *"less recently"* suggest a relation to a previously-mentioned (and in-focus) past event. As a result, we posit that *recently* on its own behaves as an abstract temporal point best annotated as a timex (as seen in the behaviour of *"until recently"* – *until* is the signal here, *recently* a TIMEX3 of value `PAST_REF`). Structures such as *[comparative] recently* may be interpreted as a qualified temporal signal, as they convey information about the relative ordering of the event that they dominate vent compared with a previously mentioned interval.

5.5.5 TB-Sig summary

Upon examination of the non-annotated instances of words that often occur as a temporal signal (such as *after*) it became evident that TimeBank's signals are under-annotated. As we are certain of some annotation errors in the source data, we revisited the original annotations. A subset of signal words was selected for re-annotation. This set consisted of signals that were ambiguous (occurred temporally close to 50% of the time) or that we expected, based on informal observations, would yield a number of missed temporal annotations. All temporal instances of these words were re-annotated with TimeML, adding `EVENT`s, `TIMEX3`s and `TLINK`s where necessary to create a signalled `TLINK`.

A single annotator checked the source documents and annotated 69 extra signals, as well as adding 34 events, 1 temporal expression and 48 extra temporal links. This left 712 SIGNALs that support `TLINK`s and 780 `TLINK`s that use a signal, with 54 signals being used by more than one `TLINK`. No events, timexes or signals were removed.

A summary of frequent candidate signal expressions is given in Table 5.14. The corpus is available via http://derczynski.com/sheffield/. Given this new, curated ground truth for temporal signal annotation, we are now ready to begin approach automatic signal annotation: firstly distinguishing temporal from non-temporal candidate expressions, and then linking signal expressions with the interval annotations that they co-ordinate.

5.6 Signal Discrimination

The words and phrases that can act as temporal signals do not always convey a temporal relation. Some may indicate possession, or a spatial relation (see Section 5.4.4). If we are to automatically annotate signals, we need to develop a method for choosing which words and phrases in a discourse are temporal signals. This task, of finding temporal signal phrases, is called temporal signal **discrimination**.

This section begins with a problem definition and description of the method we adopted to address the problem. An automatic signal discrimination technique is trained using TimeML

Expression	Count in corpus	As signal	Proportion as signals	After curation	Proportion
in	1214	161	13.3%		
after	72	56	77.8%	66	91.7%
for	621	52	8.4%		
if	65	37	56.9%		
when	62	35	56.5%	56	90.3%
on	344	33	9.6%		
until	36	25	69.4%	36	100.0%
before	33	23	69.7%	30	90.9%
by	356	20	5.6%		
from	366	19	5.2%		
since	31	17	54.8%	18	58.1%
through	69	15	21.7%		
as	271	14	5.2%		
over	59	14	23.7%		
already	32	13	40.6%	13	40.6%
ended	21	13	61.9%		
during	19	13	68.4%		
at	311	11	3.5%		
previously	19	11	57.9%	16	84.2%
within	23	8	34.8%		
s	10	8	80.0%		
later	15	7	46.7%		
earlier	50	6	12.0%		
while	39	6	15.4%	9	23.1%
then	23	5	21.7%		
once	15	5	33.3%		
still	35	4	11.4%		
following	15	4	26.7%		
meanwhile	14	4	28.6%	9	64.3%
at the same time	6	4	66.7%		
to	1600	3	0.2%		
into	63	3	4.8%		
follows	4	3	75.0%		
subsequently	3	3	100.0%		
followed	10	2	20.0%	4	40.0%
former	16	0	0.0%	12	75.0%

Table 5.14: Frequency of candidate signal expressions in TimeBank and TB-sig. We include counts of how often these occur as signal expressions both before and after manual curation.

annotations. Finally, we present results showing automatic accuracy near or above gold-standard corpus IAA.

5.6.1 Problem Definition

The temporal signal discrimination problem is as follows: Given a closed class of signal words or phrases and a discourse annotated with times and events, identify the temporal signals. This task resembles word sense disambiguation (Stevenson and Wilks, 2005; Navigli, 2009), in that given a word or phrase that may have multiple senses and its context, we have to determine if the active sense in context is a temporal one.

5.6.2 Method

The approach taken to automatic temporal signal discrimination is a supervised learning one.

We agreed a corpus and a set of words that could occur as signals. Next, we determined a set of feature variables that describe a word in context. After this we described each occurrence of a potential signal phrase in the corpus as a feature vector. Each instance was assigned a binary classification: positive if it is TimeML-annotated as a signal that is associated with a `TLINK`, or negative otherwise. Finally, we trained a classifier with these instances and evaluated its performance.

5.6.3 Discrimination Feature Extraction

As well as surface features from TimeML, syntactic features were used as part of feature extraction for signal discrimination.

Parsing and Other Syntactic Annotation

Syntactic information is likely to be of use in the signal discriminiation task . Lapata and Lascarides (2006) had some measure of success at learning a temporal relation classifier using sentences that contained signals, with syntactic information as a core part of their feature set. Their work used the BLLIP corpus,[6] which contains around 30 million words from Wall Street Journal articles and constituent parses generated by the Charniak parser (Charniak, 2000).

To attempt to partially replicate this source information, we parsed the text of the TimeBank corpus. Note that TB-sig and TimeBank differ only in the annotations that they make over text; the actual words in both corpora are the same, and in the same order. To do this, we removed markup from each document and separated the remaining discourse into sentences using the Punkt sentence tokeniser (Kiss and Strunk, 2006), as part of CAVaT preprocessing (Derczynski and Gaizauskas, 2010a). Each sentence was then word-tokenised using NLTK's treebank tokeniser.[7] To maintain word alignment consistency with the non-parsed text stored in CAVaT, we needed a parser that accepted external tokenisation. We chose the Stanford parser (Klein and Manning, 2003) for generation of constituent parses.

[6] LDC catalogue number LDC2000T43

[7] See http://www.nltk.org/ for more information on this package.

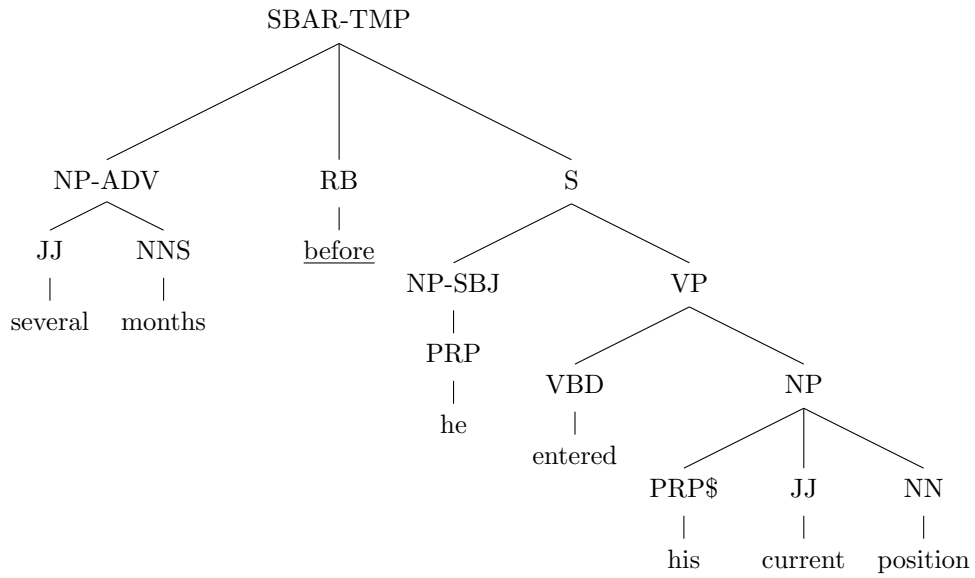

Figure 5.6: Example of an SBAR-TMP where the first child is a signal qualifier (*several months*) and the second child the signal word itself (*before*).

In addition to constituent parses, the BLLIP corpus includes **function tags**. These are optional labels (Marcus et al., 1994) attached to nodes in a constituent tree. Function tags extend a constituent tag by providing additional information about the role it plays in a sentence. They exist in three main groups; syntactic, semantic and topical (Bies et al., 1995). Of direct interest to us is the `-TMP` tag, which indicates temporal function. An example of this tag is given in Figure 5.6, where the first children of an `SBAR-TMP` node comprise a temporal signal.

Early work on function tag assignment in conjunction with the Charniak parser was performed by Blaheta and Charniak (Blaheta and Charniak, 2000). Their approach found that choosing whether or not to assign any tag was a significant and difficult component of the task. Thus, evaluations are split into "with-null" and "no-null" figures, where with-null refers to tag assignment accuracy including the assignment of no tag to untagged constituents and no-null is the proportion of correctly-tagged constituents excluding non-tagged nodes. We refer to no-null performance figures when discussing taggers. The initial Blaheta tagger had an F-measure of 67.8% on the semantic form/function category, which includes the `TMP` tag.

We would like to use a function tagger with good `TMP` tagging performance. Musillo and Merlo (2005) simultaneously parsed and tagged text using a Simple Synchrony Parser and an extended tag set. This generated lower results than Blaheta's original attempt though this was improved to provide a marginal increase using input sentences annotated by an SVM tagger. Blaheta's final tagger (2004) improved semantic tagging to 83.4% F-measure, which was comparable to later work in which overall tagging performance increased (Gabbard et al., 2006; Lintean and Rus, 2007). As the final Blaheta tagger is freely available and openly distributed, we used this to augment our constituency parser (the Stanford parser (Klein and Manning, 2003)).

We only treated as positive examples signals that were associated with a `TLINK`. Signals that only provided information regarding event cardinality, or to subordinate or aspectual links, were ignored. Signals with text not in our closed class of signal words and phrases were ignored.

Basic feature set

Our initial features were both syntactic and lexical; a list of them is given below. Lexical and TimeML-based features were extracted directly from a CAVaT database constructed from TimeBank (Derczynski and Gaizauskas, 2010a). We use NLTK's built-in Maximum Entropy classifier.

a. Part-of-speech from PTB tagset (Marcus et al., 1993). (`sig_pos`)

b. Function tag from Blaheta tagger; if there is more than one and the set includes TMP, assign TMP, otherwise assign the first listed. (`sig_ftag`)

c. Constituent label and function tag of parent node in parse tree (two features). (`parent_pos`, `parent_ftag`)

d. Constituent label and function tag of grandparent node in parse tree (two features). (`gparent_pos`, `gparent_ftag`)

e. Is there any node with the TMP function tag between this token and the parse tree root? (`tmplabel_in_path`)

f. Signal text. (`text`)

g. Text of next token in sentence (if there is one). (`next_token`)

h. Text of previous token in sentence (if there is one). (`previous_token`)

i. Is there a `TIMEX3` in the n following tokens? (`timex_in_n_after`)

j. Is there an `EVENT` in the n following tokens? (`event_in_n_after`)

k. Is there a `TIMEX3` in the n preceding tokens? (`timex_in_n_before`)

l. Is there an `EVENT` in the n preceding tokens? (`event_in_n_before`)

m. The Stanford dependency relation of the candidate word to its parent. ()

In our work, $n = 2$ for the interval proximity features, based on an informed guess after looking at the data. The optimal value, depending on direction of context and type of interval (event vs. timex) search for, is left to future work.

There are 102 entries in our closed class of signal words/phrases; this set is kept constant throughout all experiments. In TimeBank there are 7 014 mentions of the members of this set, including both temporal and non-temporal mentions.

Measure	Accuracy	Accuracy (+ve)	Error reduction
Extended features			
Naïve Bayes	86.1	80.9	-4.28
Maximum Entropy	89.4	55.4	19.9
ID3	89.9	59.8	24.2
C4.5	90.4	60.8	27.8
AdaBoost	**90.6**	**59.3**	**29.3**

Table 5.15: Signal discrimination performance on the TimeBank corpus, with an extended feature set. Error reduction is measured relative to most-common-class ("not a signal") performance. Evaluated with 5-fold cross validation and 1 000 iterations of adaptive boosting.

Extended feature set

Curation of signals, as detailed in Section 5.5 above, led to some direct observations about specific signal words. These observations in some cases suggested specific sources of signal discrimination information thar could potentially be translated to features. From the observations above, the new features that could be added were:

n. Flag to see if signal text is in a verb group (*before, after*) (`in_verb_group`)

o. Flag to see if a token at the top of a paragraph (*previously*)

p. Flags to see if the preceding or following word(s) are part of a verb group (*after*) (`following / preceding_in_verb_group`)

q. What is the highest-level subtree that begins at the next token (*before*) (`following_subtree`)

r. What is the highest-level subtree that ends at the preceding token (`preceding_subtree`)

s. PoS of the next token and previous token (*before*, *after*) (`following/preceding_pos`)

t. PoS of the next event within *n* tokens (*before*, *former*) (`next_event_pos`)

u. Type (TimeML `class`) of the next event within *n* tokens (*former*, *meanwhile*) (`next_event_class`)

v. TimeML Tense and aspect of the next event within n tokens (*already*) (`next_event_tense / aspect`)

w. NP begins at next token? (*former*) (`np_next`)

x. Is the preceding token a comparative, i.e., is it one of JJR or RBR? (*recently*) (`preceding_comparative`)

All of these were implemented and added as features, except the paragraph-top feature (due to a lack of a reliable document segmentation tool). In addition, we removed some noisy features that seemed to be causing overfitting within our sparse data set; the offset of the word within its

Features	NBayes	MaxEnt	ID3
Full subtree labels	-1.32	19.4	**25.4**
Just constituent tag	-2.31	19.7	**21.6**
Separate constituent and function tag	-4.28	19.9	**24.2**

Table 5.16: Comparison of the effect that decomposing values of the preceding_subtree and following_subtree features has, using our extended feature set and TimeBank data. Error reduction compared to classifier MCC baseline.

sentence and the preceding & following token texts. We used the full constituent tag of subtrees for the `preceding_subtree` and `following_subtree` features, including .

Multivalent tags

In a minority of cases, constituents and terminals were assigned multiple function tags. For example, values such as PRD-TPC-NOM or TMP-SBJ would be appended. Noticing that these instances were assigned high weights by a Naïve Bayes classifier, we measured error reduction on multiple variations of subtree tag feature representations. Results are shown in Table 5.16. It was found that reducing data sparsity by providing two separate features per subtree (for constituent tag and function tag) provided best overall performance for MaxEnt discriminators, but ID3 benefited most from the feature extraction that gave the sparsest values – full subtree labels.

Choice of learning algorithm

Signal discrimination is a binary classification problem: is a given word or phrase a temporal signal or not? We have constrained the set of words we attempt to classify by defining a closed class of signal words and described a set of features with which we will represent candidate words and context. We now need to choose a binary classification algorithm. We use a Naïve Bayes classifier, decision trees, a maximum-entropy classifier and adaptive boosting.

For rapid learning and quick feedback, we worked with the Naïve Bayes classifier. Naïve Bayes models are computationally cheap to learn. Its inductive bias includes the independence assumption – that all features are independent from each other. This is not true in our case, given the heavily interdependent nature of most of our features: well-formed syntactic structures are inherently constrained by grammar and the values of many of our features depend on syntax at multiple places in the same sentence or paragraph. For example, the parts of speech of any given token has some bearing on the part of speech of the following one, and these are again not independent of the parse tree of the sentence in which they occur. We also use a decision tree classifiers, which do not have this particular bias and are computationally quick to learn, but do not always cope well with noise. ID3 and C4.5 types are used. C4.5 attempts to deal with noise in training data by performing pruning on the tree after construction (Quinlan, 1993).

We also evaluate performance of our feature set with a maximum entropy classifier. This regression-based model assumes low collinearity between features, which is a less constraining

Baseline	Accuracy	Accuracy on positives
Most common class	86.7%	0.0%
Baseline: Part-of-speech is IN	25.6%	**81.2%**
Baseline: Part-of-speech is WRB	86.9%	5.77%
Baseline: Parent is SBAR-TMP	**87.0%**	9.88%
Baseline: Parent function is -TMP	*8*4.5%	72.7%

Table 5.17: Performance of four constituent-tag based baselines over TimeBank.

assumption than that of the Naïve Bayes classifier, though problems may arise if we use highly-correlated features. Finally, we use adaptive boosting with decision stumps (Freund and Schapire, 1997, 1996), which is constrained to binary classification and can yield high-performance results. Adaptive boosting reduces the impact of the typically computationally intensive SVM-learning process and typically displays little overfitting, which is helpful with smaller datasets such as ours.

Performance was improved by removing features that have a high number of values (for example, the text of the token after a signal). We suspect this is due to them leading to overfitting.

5.6.4 Discrimination Evaluation

We have described how we trained a classifier using cross-validation. We evaluated performance using a held-out evaluation set, and determined scores by counting correct classifications and measuring both percentage of correctly classified instances and also the error-reduction compared to a baseline.

Baselines

To evaluate the performance of our approaches, it is useful to describe some simple annotation methods as baselines. A summary of our baselines is given in Table 5.17 and we explain each of them below.

One simple baseline is to find the most common classification and assign this to all instances. In our corpus, instances of phrases from our list of potential signals are used non-temporally nearly all the time (out of 6 091 instances of potential signal phrases, only 688 are annotated as being temporal signals in TimeBank – 11.3%) and so our most common case is to classify everything as not being a temporal signal, regardless of the signal text.

We also use baselines that mark all words found in the signal phrase list as temporal signals if they have a part-of-speech tag of RB or IN, according to NLTK's built-in maximum entropy tagger. Values are quoted for overall classification accuracy, as well as accuracy on positive examples (the minority of our training data).

Most common class The training set is confined to just signal annotations in TimeBank/TB-sig, that are also in the closed class of signal expressions detailed above in Table 5.9. This introduces an inherent performance cap to the overall approach, but assumes no knowledge of whichever corpus is being used as the evaluation set. Of 4 576 training instances, 3 969 are negative (non-temporal) and 607 are positive (having a temporal meaning). The most-common-class is negative and if we assign this label to all mentions of members of the set, classifier accuracy is 86.7% but no signals are identified (giving an effective F1 of zero if we imagine this as a signal recognition task); not a very informative baseline.

Class member and signal word tag Of all leaf labels, IN and WRB have the highest proportion of signals (Table 5.10). To this end, we have two simple baselines, where we count a word as a temporal signal if its constituent tag is IN or WRB and it is found in the closed class of signals. Performance for these is given in Table 5.17. For IN, we have 25.6% overall accuracy, correctly identifying text that is a temporal signal 81.2% of the time. For WRB, we achieve 86.9% accuracy, but only 5.77% on the positive examples.

Parent is SBAR-TMP As mentioned in Section 5.4.2, one might expect an a SBAR-TMP subtree to begin with a temporal signal and also contain one of the signal's arguments (see also Figure 5.6). As we can use our closed class of signal words to differentiate signal head, signal qualifier and event/timex argument, we can look for leaves where the parent is SBAR with TMP in its function tags. This is our SBAR-TMP baseline, that performs at 87.0% accuracy overall, with 9.88% on positives – better than WRB, but still poor.

Parent has temporal function Limiting ourselves to just signals in subtrees labelled SBAR may be a short-sighted manoeuvre. We added a baseline that labels signal candidates as temporal if their parent has a temporal function label. This baseline achieves classification accuracy of 84.5% and a 72.7% accuracy on the positive examples; see Table 5.17.

Performance

With our original feature set and based on pre-curation data (e.g. TimeBank v1.2), we achieved a 40% error reduction in signal discrimination relative to a competitive baseline, as seen in Table 5.18. For the general annotation task, naïve Bayes performed best, with good error reduction overall (26.5%) and a similar improvement in recognition of positive examples (20.9%), something that other classifiers did not perform so well with.

With the original feature set, models learned over TB-sig data performed as shown in Table 5.19. Performance using the extended feature set is detailed in Table 5.15, again based on TB-sig.

Our extra annotations introduce new signal instances for the extra terms that we have annotated, reducing the baseline to 85.2% accuracy (677 positives, compared to 607 before re-annotation) from 86.7% before – see Table 5.19. Performance using TB-sig is overall better (compared to Table 5.18), which we attribute to having a better-stated problem and less mislead-

Measure	Accuracy	Accuracy (+ve)	Error reduction	Error reduction (+ve)
Naïve Bayes	88.6	**78.4**	26.5	**20.9**
Maximum Entropy	89.5	56.0	32.3	-61.2
ID3	90.5	65.6	38.7	-26.0
C4.5	90.4	60.1	38.1	-46.2
AdaBoost	**90.7**	59.8	**40.0**	-47.3

Table 5.18: Signal discrimination performance on the plain TimeBank corpus. Error reduction is measured relative to the "parent has temporal function" baseline. Evaluated with 5-fold cross validation and 1 000 iterations of adaptive boosting.

Measure	Accuracy	Acc. (+ve)	Error reduc.	Error reduc. (+ve)
Most common class	85.2	0	n/a	n/a
Baseline: IN	25.4	77.1	-	-
Baseline: RB	86.3	8.3	-	-
Baseline: SBAR-TMP	86.1	10.8	-	-
Baseline: Temporal parent	84.5	70.0	-	-
Simple features				
Naïve Bayes	89.3	**78.7**	31.0	29.0
Maximum Entropy	88.2	51.3	23.9	-62.3
ID3	91.7	69.6	46.5	-1.3
C4.5	**92.1**	73.0	49.0	10.0
AdaBoost	91.9	70.5	47.7	1.7
Extended features				
Naïve Bayes	87.0	**81.4**	16.1	38.0
Maximum Entropy	88.1	50.1	23.2	-66.3
ID3	91.1	68.7	42.6	-4.3
C4.5	91.7	75.0	46.5	16.7
AdaBoost	**91.8**	69.3	47.1	-2.3

Table 5.19: Signal discrimination performance on the curated corpus. Error reduction is measured relative to performance. Results are for 5-fold cross validation. Adaptive boosting used 1 000 iterations.

ing data. Error reduction rate is now over 40%, with overall accuracy just under 92% and up to 75% on the positive examples. This is better than performance on the original TimeBank data and comparable to the IAA figure of 0.77 for TimeBank's initial `SIGNAL` annotation. C4.5 performs particularly well, reaching near-highest error reduction rate and good accuracy on positive examples.

The extended feature set, however, does not improve performance in the majority of cases, despite having been generated as part of a rational investigation. Analysis and further work is required to improve upon these signal discrimination results.

Useful features

A sample post-classification analysis of feature weights – using TB-sig and the extended feature set – is presented in Table 5.20, taken from the last of five cross-validation passes. This is from the construction of a model using the whole signal-labelled corpus with a naïve Bayes classifier. The text of the signal is a particularly strong indicator for some of the features that occur much more often as temporal signals than not. We can also see that wh-adverb signals and wh-adverb phrases that contain the candidate signal expression are strong indicators of temporal meanings (features signal_label, parent_label and ending_subtree_label); this may be because of words such as *when* having only temporal senses. A timex or a past-tensed event occurring after the signal is also an indicator of it being temporal (timex_in_2_after). When the parent constituent or the largest constituent beginning at this point has a temporal function, then a candidate word is more likely to be temporal (parent_function, starting_subtree_function). The -TMP function tag helps to indicate a temporal signal when it dominates the candidate signal word (tmpfunction_in_path). Being followed by a dollar amount suggests that a candidate is not temporal (following_label = $) – for example, in a non-temporal use, *"Shares closed at $ 50"*; the high weight of this attribute-value pair is likely influenced by the high proportion of financial reporting in TimeBank, which takes a significant part of its text from the Wall Street Journal.

Words and phrases that are within a syntactical structure that has a spatial function (e.g. -LOC) contra-indicate a temporal meaning. This is aligned with the observation that members of our class of signal words often have both temporal and spatial meanings. Further, an adjacent structure with a spatial function (-EXT or -LOC) suggests a temporal function in a candidate word. This suggests collocation based approaches may not correctly discriminate temporal and non-temporal signals; syntactic parsing is required, in order to detect these functional nuances. Having NX (indicating the head of a complex NP) as a parent at can indicate a signal; this could be in cases where we have a signal before a nominalised event, such as in *"before the explosion"*. Finally, preceding a verb may be an indication of a temporal signal; this reflects the signal's adverbial nature.

5.6.5 Discrimination on unseen data

Up to this point, evaluation has used cross-validation over TimeBank. Our error analysis led to the inclusion of features based on the data that is also part of the evaluation set. To check performance on previously unseen data, a further experiment was performed is as follows. We trained a signal

Feature	Value	Indication	Weight
text	until	True	131.5
text	before	True	70.0
text	after	True	56.9
signal_label	WRB	True	49.6
parent_label	WHADVP	True	49.5
ending_subtree_label	WRB	True	48.5
text	when	True	48.3
text	previously	True	26.2
text	former	True	15.4
grandparent_label	SBAR	True	13.9
text	during	True	11.5
following_subtree_function	-LGS	False	9.7
text	meanwhile	True	9.6
timex_in_2_after	True	True	9.0
text	since	True	7.6
preceding_subtree_label	S	True	7.2
starting_subtree_function	-LOC	False	7.1
following_label	$	False	7.0
starting_subtree_label	SBAR	True	6.6
parent_function	-LOC	False	6.4
following_subtree_label	VBN	True	6.3
starting_subtree_function	-TMP	True	6.2
following_label	PRP	True	6.1
grandparent_label	NX	True	5.7
starting_subtree_label	NX	True	5.7
preceding_label	JJS	True	5.6
following_subtree_label	VB	True	5.6
text	thereafter	True	5.6
next_event_tense	PAST	True	5.4
parent_function	-TMP	True	5.3
parent_label	SBAR	True	5.3
text	later	True	4.9
tmpfunction_in_path	True	True	4.1
preceding_subtree_function	-LGS	True	4.1
preceding_subtree_function	-EXT	True	4.1
following_subtree_function	-PRD	True	4.1
starting_subtree_function	-TPC	True	4.1
grandparent_label	SINV	True	4.1
following_subtree_label	.	False	4.0
following_label	.	False	4.0

Table 5.20: Sample features useful for signal discrimination, based on our curated TimeBank data, TB-sig.

Feature	Pre-curation	Post-curation
Documents	15	
Tokens	7 099	
Signals	96	114
TLINKs	1048	1062
Events	1060	1060
Timexes	154	156

Table 5.21: Characteristics of the N45 section of the AQUAINT TimeML corpus, before and after signal curation

Method	Accuracy	Precision	Recall / acc. on positives
Parent `-TMP` baseline	84.5%	-	70.0%
MaxEnt model	93.6%	83.0%	78.3%

Table 5.22: Performance of a TB-sig trained signal discriminator on unseen data

discriminator and associator based on all of TimeBank + the extra signal annotations. The closed class is increased to include all phrases marked as signals in TimeBank. This way, TimeBank is only the training data.

As the final model was developed based partially on observations of TimeBank, it is not suitable to evaluate the final model on this corpus also. A previously unseen set, taken from the AQUAINT corpus (Section B.2.2), now forms the evaluation set. The N45 section of the AQUAINT corpus was curated to verify its signal annotations, and then signal discrimination was evaluated over this subcorpus based on a model trained on the entirety of TB-sig. The relevant statistics regarding this evaluation corpus are presented in Table 5.21.

Signal discrimination is measured in two ways. Firstly, classification accuracy shows how many of the candidate signal words were correctly labelled as signals or not-signals. Secondly, the overall performance of the association approach at annotating signals in any given document is described in terms of precision and recall. This takes into account how well the entire approach described above (including the signal words list described in Table 5.9, but not also including those found in TimeBank) does when given the task of identifying temporal signals in an arbitrary text. The augmented AQ/N45 annotations form the gold standard. The "parent has temporal function" baseline (Section 5.6.4) is used for comparison. Results are presented in Table 5.22. This compares well with the performance on (seen) TB-sig data (Table 5.18).

5.6.6 Summary

In this section, we have explored the task of signal discrimination. We discovered that TimeBank's signal annotations are incomplete. To remedy this, we have proposed augmentations to the TimeML annotation standards and re-annotated a portion of the corpus. We have also defined

a set of features that can describe a temporal signal in context and constrained our search space to just words and phrases in a closed class of signal words. As a result, we have been able to train a classifier to detect temporal signals at near-IAA accuracy.

5.7 Signal Association

Temporal signals connect one or more interval pairs and describe the nature of the temporal relation between the pair. This section describes an investigation into how to find the arguments of a temporal signal, thus associating the two arguments. We refer to this task as**signal association**.

In order to fully annotate temporal signals, we need to determine which arguments they co-ordinate. To this end, the task of determining which times or events are coordinated by a temporal signal is examined as the subject of this section.

5.7.1 Problem definition

When performing temporal annotation, one needs to identify events and times and can then connect them with temporal links, perhaps using an associated signal. In fact, every time that a temporal signal is annotated, there must be a temporal link present. The signal association problem is: Given text with signal, event and timex annotations, determine which pair of events/times are associated by each signal phrase.

5.7.2 Method

A supervised learning approach is taken to finding which intervals a given signal co-ordinates. TB-Sig is used as the dataset for feature extraction. Two approaches are explored, detailed below. These use a largely common feature set, extracting a number of features for each interval considered and a further set of features describing the signal.

To generate training data given a signal, we will describe events and timexes within the scope of that signal using our feature set. Although any two intervals in a document could be linked by a given signal, the number of intervals or interval pairings one must search through could be large if the entire document is used as potential signal scope. For this reason, scope must be constrained, at a possible performance loss. Given candid examination of the signals in the corpus, the scope of the signal is taken to be the signal's sentence and also enough previous sentences to include at least two intervals, as well as a DCT timex if present. We are attempting to determine which intervals are associated with the signal.

The goal is to learn a binary function, that can indicate whether or not an association supporting a `TLINK` exists in a given situation. A TLINK associates two intervals (timex or event) and may specify the type of temporal relation between them. We have tried two approaches to this signal association task; one where we examine ⟨interval, signal⟩ tuples and another where we examine ⟨interval-pair, signal⟩ tuples. The gold standard corpus, TimeBank, provides the positive examples. For each signal, there may be up to five valid `TLINK`s, each shown as an interval pair (see earlier Table 5.8).

For the single interval approach, we train a binary classifier to learn if an interval and signal are linked and then choose the two best candidate intervals for a signal, using classifier confidence to rank similarly-classified intervals. For the interval pair approach, for each signal we examine possible combinations of intervals and create a vector of features based on relations between the intervals and the given signal.

Single Interval Approach

In this section, we describe a signal association approach where individual intervals are ranked by their relation to the signal and the top two intervals are deemed to be associated.

Positive training examples came from intervals associated in a gold standard annotation. Negative training examples were taken to be all temporal intervals in the same sentence as the signal that were not associated with the signal. We used cross-validation to learn classifiers and recorded the prediction and confidence of the classifier for each entry in the evaluation fold. After this, for each signal, a list of candidate intervals was determined. The two intervals related to the signal were those classified as related with highest classifier confidence, or if fewer than two positive classifications were made, up to two are taken from lowest-confidence unrelated classifications. That is, for each signal, intervals are ranked in descending order of confidence; the goal is to find the two most likely intervals, and associate them in a TLINK backed by the given signal. Priority is established in this order:

1. High-confidence and classified as related
2. Low-confidence and classified as related
3. Low-confidence and classified as unrelated
4. High-confidence and classified as unrelated

The top two are then associated with a signal. This approach is limited to only detect one pair of intervals per signal.

Interval Pair Approach

In contrast to our previous approach, we tried to identify whole ⟨interval-pair, signal ⟩ 3-tuples as either a signalled `TLINK` or not. This produced a majority of negative examples. We instead only considered intervals where both arguments fell inside a sliding window of sentences, to reduce the heavy skew in training data. A boolean feature describing whether the intervals were in the same sentence was added to our set, as well as two sets of interval-signal relation features and general signal features as described earlier.

Surface and Constituent-Parse Features

For the signal association tasks, we used the following surface and constituent-parse features as input to a binary classifier. Constituent parse information comes from running the Stanford Parser (Klein and Manning, 2003) over discourse sentences, the bounds of which are determined using the Punkt tokeniser (Kiss and Strunk, 2006) implementation in NLTK. The features describe

a single interval/signal pair. We use the same definition of syntactic dominance as (Lapata and Lascarides, 2006); that is, an interval (e.g. event or timex) is syntactically dominated by a signal if the interval's annotated lexicalisation is found within a parse subtree where the first (leftmost) word of the parse subtree is the signal. Dominance features are included based on their success in signal linking in Lapata and Lascarides (2006), where dominance was described as the V_L feature.

- Is this interval the textually nearest after the signal?
- Is this interval the textually nearest before the signal?
- Does the signal syntactically dominate the interval?
- Signal text (lower case)
- Signal part of speech
- Token distance of interval from signal
- Interval/signal textual order
- Is there a comma between the interval and signal?
- Is the interval in the same sentence as the signal?
- Is the interval DCT or a DCT reference?
- Interval type (TimeML `EVENT` class or `TIMEX3` type), total 11 values
- If an event, its TimeML-annotated tense

Dependency Parse Features

We use the Stanford dependency parser (De Marneffe et al., 2006) to return dependency graphs of our PoS-tagged, parsed and function labelled sentences. By default, the dependency parser ignores some words that we consider to be signal words, moving information about removed words in relationships. We configured it to never ignore words. The features that we extracted from sentence dependency parses were:

- Length of path from interval to root
- Is the signal a child of the interval?
- Is the signal a direct parent of the interval?
- Is the interval the tree root? (e.g., the head event/time)
- Is the interval directly related to the signal with an `advmod` or `advcl` relation?
- Does the interval modify the root directly? (e.g., is the interval a direct ancestor of the root, regardless of relation type)
- Does the signal modify the interval directly? (e.g., is the signal a direct ancestor of the interval)
- What relation does the interval have to its parent?
- If the signal is a child of the interval, what is the relationship type?

5.7.3 Dataset

Examining some of the instances of temporal relations in TimeBank which have an attached signal, there were often clear syntactic relations between signals and their arguments (which are also the

Distance	Count
DCT	40
0	682
1	43
2	16
3	3
4	3
5+	0

Table 5.23: Distribution of sentence distance between intervals linked by a signal, for TB-sig. A special case is made for those that link to document creation time or one of its co-referents, as it often persists as a reference point through the length of a discourse.

temporal relation's arguments). Almost all signals co-ordinated two intervals in the same sentence as the signal (Table 5.23). In the cases where they did not, one of three situations prevailed. Firstly, the signal was the first token in the sentence and the argument outside of the sentence was either referenced by a temporal pronoun (as in e.g. *"After **that**, the situation improved."*). Secondly, one argument is an event or time that has remained the temporal focus in discourse at the point where the signal is found, even after new sentences have been introduced. Thirdly, the signal will relate DCT with an interval in its sentence.

Closure

Some supervised approaches that deal with temporal relations chose to use closure to generate extra training data. We have deliberately chosen not to include temporal links generated through closure (Verhagen, 2004) in our examples. Temporal closure typically generates more links than were in the original annotation by at least an order of magnitude. The generated links tend to be between intervals not directly related in text – e.g. lacking textual proximity or clear discourse relations. As with many binary classification models, the negative examples that enable our classifiers to learn the most precise decision boundaries are those that closely resemble positives. Entities only linked through a chain of four or five annotated `TLINK`s, with low textual or syntactic proximity, will not be in this set. We do however use windowing approaches to permit some of these wide-ranging negative examples into the training.

Detecting Document Creation Time

Document creation time (**DCT**) refers to the instant at which a discourse was created. In the case of newswire articles this is often included in the article metadata, or as a deictic temporal expression at the beginning of the first sentence, which describes day and month (e.g. *"KABUL, August 21 – ..."*). The document creation time persists throughout a discourse as an antecedent temporal point that may be referred to by temporal expressions or, in some cases, signals. As we have seen some signals that work like this (e.g. *afterwards*), it may be useful to include a boolean

feature indicating whether or not a timex represents DCT.

TimeML-annotated data is used to determine whether a given timex is DCT or DCT-equivalent. Our algorithm is as follows, given a candidate `TIMEX3` element:

1. if `functionInDocument` = `CREATION_TIME` ⇒ return **true**
2. if `functionInDocument` = `PUBLICATION_TIME` ⇒ return **true**
3. `most-frequent-anchor` ← the most frequent non-null value of `anchorTimeID` in this document's `TIMEX3` annotations
4. if `sentence-number` $< j$ and `timex_id` = `most_frequent_anchor` ⇒ return **true**
5. else return **false**

That is, we first look for explicit annotation markers that declare this timex to be a creation time reference. Failing that, if the timex is near the beginning of the document and also the timex most-often used as an anchoring point for other timexes, we mark it as DCT-referring. With $j = 2$, this heuristic is accurate for all of TimeBank.

5.7.4 Automatic association evaluation

As both approaches rely on a binary classifier, the first evaluation measure given is classifier accuracy. This shows the proportion of accurate binary decisions made by the classifier based on model learned from training data. The error reduction that the classifier's model provides over a most-common-class baseline is also given. The single-interval approach and interval-pair approaches are structurally different and can be further evaluated in separate ways, which are detailed below, as well as results.

Single-interval

We recognised three possible states of signal annotation. A **full match** occurs when both signal arguments are correctly found, when just one argument is correct we have a **partial match** and when both associated arguments are incorrect there is a **failure**. Results of classifier performance and signal annotation success can be found in Table 5.24. Full matches are the only cases we should consider as successes; anything else is not correct, though partial successes (where one argument is correctly associated) are shown to give insight into how problematic the non-full matches were. As can be seen from the data, even in cases where there was not a full argument match, it was almost always the case that at least one interval was correctly associated – that is to say, partial matches were orders of magnitude more common than failures.

Interval-pair

Results for the interval-pair:signal approach are given in Table 5.25. The "Acc (+ve)" column represents the classifier accuracy on examples labelled as positive in the gold standard, as opposed

Corpus	Classifier	Accuracy	Err. reduc	Full	Partial	Failure
TimeBank	MaxEnt	**85.2**	58.7	**64.2%**	34.5%	1.25%
	NBayes	82.5	51.1	57.2%	41.2%	1.53%
	ID3	78.4	39.8	42.1%	52.1%	5.85%
TB-sig	MaxEnt	**84.8**	57.9	**61.5%**	37.6%	0.897%
	NBayes	82.2	50.5	56.3%	41.9%	1.79%
	ID3	79.6	43.4	40.9%	54.4%	4.74%

Table 5.24: Performance at the signal:interval association task, with 5-fold cross validation. The classifier performance baseline is most-common-class, which was 64.1% not-related for TimeBank and 64.0% not-related for the signal-augmented version.

Corpus	Classifier	Accuracy	Err. reduction	Acc. (+ve)
TimeBank n=0, baseline 89.6	NBayes	94.0	41.8%	91.4
	ID3	97.7	77.3%	84.7
	MaxEnt	92.5	28.0%	43.7
TimeBank n=1, baseline 96.6	NBayes	93.6	-89.4%	93.9
	ID3	99.3	**79.9%**	84.0
	MaxEnt	97.1	13.9%	43.6
TimeBank n=2, baseline 98.3	NBayes	94.7	-219%	**95.5**
	ID3	**99.4**	62.1%	68.7
	MaxEnt	84.9	-804%	39.3
TB-sig n=0, baseline 89.7	NBayes	94.1	42.8%	90.8
	ID3	97.4	74.8%	84.8
	MaxEnt	92.2	23.6%	41.6
TB-sig n=1, baseline 96.7	NBayes	93.4	-100%	93.2
	ID3	99.3	**78.0%**	83.5i
	MaxEnt	97.1	12.3%	44.5
TB-sig n=2, baseline 98.4	NBayes	94.7	-229%	**94.7**
	ID3	**99.1**	42.7%	46.8
	MaxEnt	84.9	-832%	38.8

Table 5.25: Performance at the signal:interval-pair association task, with 5-fold cross validation. The baseline is most-common-class, which was "no link" in all cases. The sentence window for negative examples is the signal's sentence plus the n prior sentences.

	Prediction	
Class	True	False
True	564	872
False	12110	72192

Table 5.26: Confusion matrix for signal association performance with a MaxEnt classifier on TimeBank with a window including the signal sentence and two preceding ones.

Distance	**Count**
DCT	41
0	1468
1	43
2	16
3	3
4	3
5+	0

Table 5.27: Distribution of sentence distance between intervals and signal that links them. A special case is made for those that link to document creation time or one of its co-referents, as in Table 5.23.

to the proportion of the instances labelled as positive that were matched the gold standard annotations. The best classifiers are those that achieve a high error reduction while maintaining good classification accuracy on positive examples.

For most Naïve Bayes classifier results, there were was a low false negative and a high true positive rate, but also an overbearing false positive rate. For example, with $n = 2$ there were 1371 true positives and only 65 false negatives, which is good, but 4513 false positives, meaning that the classifier output was not particularly useful. Less than one quarter of interval-pair:signal associations would be accurate. Table 5.26 shows the confusion matrix of the worst-performing attempt. It detects a large number of false positives.

Using windowing for candidate interval selection with $n = 2$, 0.38% of signal arguments lie out of the window (see Table 5.27) and are therefore not correctly associable with this approach – an acceptably small amount. With $n = 0$, this unassociable proportion rises to 4.13%. We found that increasing n led to worse classifier performance and a value of $n = 1$ provided a good trade-off.

Performance is worst with $n = 2$. We can achieve a good classification accuracy on a test set that includes cross-sentence links even if we only consider same-sentence intervals for the generation of negative examples (i.e. $n = 0$). We can also see that decision trees, which do not follow the independence assumption, perform consistently well, although do worse as n increases.

Evaluating on previously unseen data

To test association on its own, a classifier is trained on TB-sig and evaluated on the augmented AQ/N45 data (a TimeML subcorpus introduced in Section 5.6.5). The interval pair annotation method is used, as it performs best on prior TimeML data (Section 5.7.4). The results are shown

Method	Accuracy	Error reduction	Acc. on positives
Most common class (not related)	91.96%	-	0.00%
ID model (n=1)	96.60%	57.72%	84.93%

Table 5.28: Performing of a TB-sig trained signal associator on unseen data

in Table 5.28.

This is satisfactory performance, with a strong error reduction of 58% beyond the baseline.

5.7.5 Association summary

Our aim was to find a method of automatically associating a temporal signal with a pair of intervals, given a partially annotated text. We tried two approaches. The first ranked ⟨interval, signal⟩ tuples and treated the top two as linked. The second treated ⟨interval-pair, signal⟩ tuples as atomic units.

It is important to achieve a good error reduction rate and also to have good predictive accuracy on positive examples. Both of these metrics need to have high values for a classifier to be useful in annotation. We found that although the ranked single-interval approach achieved decent results, treating interval pairs as atomic units worked better. We achieved 78.0% error reduction over the most-common-class baseline, at 96.7% predictive accuracy and 83.5% accuracy on the positive examples.

5.8 Overall Signal Annotation

The overall motivation for signal extraction is to improve automatic temporal relation typing. We have independently determined that signals are useful for TLINK typing (Section 5.3 above) and that we can extract and associate signals automatically (Sections 5.6 and 5.7). To show that automatic extraction is useful in support of the relation typing task, we took a gold-standard TimeML corpus (the AQUAINT TimeML corpus) and removed all its signal annotations. Performance of an automatic TLINK labeller was then compared when there are no signal annotations and when signal annotations have been automatically added using the above methods.

The same unseen corpus (a signal-augmented version of the N45 section of AQUAINT TimeML corpus) was used for evaluation of discrimination and association, as introduced in Section 5.6.5.

5.8.1 Joint Annotation Task

To measure combined performance, the signal annotations suggested in the discrimination step are used as the basis for association. Note that because the set of TLINKs identified in a document's annotation may not be a temporal closure of that document (see Section 3.3.2), it is possible to correctly detect a pair of events that are in fact linked via a signal but for the TLINK not to be present in the gold standard. For this reason, the performance scores are minimums. We

Signal/TLINK associations	Count	Proportion
In N45	136	-
Found	336	-
Found, both args in N45	88	26.2%
Signal in N45, new TLINK assoc	216	64.3%
Found based on new signals	32	9.5%

Table 5.29: Details of the joint approach to signal annotation. Although the augmented N45 corpus only contained 136 signals, our approach found 424. This table breaks down that 424.

hypothesise that despite a lack of guidance regarding which TLINKs must be defined in order to create a complete or valid TimeML annotation, annotators are likely to add explicit TLINK annotations where the temporal relation is suggested explicitly (e.g. with a signal). Therefore the number of unannotated signalled TLINKs should be small.

The corpus used was the augmented N45 dataset, stripped of TLINK and SIGNAL annotations (leaving TIMEX3s and EVENTs). The method was to first attempt automatic signal discrimination over the corpus (training on all of TB-sig using the basic feature set), and then perform automatic signal association (using the interval-pair approach). The resulting SIGNAL and TLINK annotations were then compared to the augmented N45 annotations.

Results are summarised in Table 5.29. In total, compared to the 136 signalled TLINKs in the augmented AQ/N45 data, 336 interval pairs (e.g. TLINK suggestions) were suggested based on the automatically annotated signals. A total of 64.7% of the 136 TLINKs were found correctly automatically. Only 26.2% of associated interval pairs (88 out of 424) were found in the gold standard; 248 were not there. A minority of 9.5% (32) of pairs found were based on signals not in the gold standard. This leaves 64.3% (216) automatically generated instances of signal associations with interval pairs not mentioned in the gold standard.

Upon manual inspection, many of these false positives based on existing signals appear to be supported in the text, but are not annotated in the gold standard, which in many cases contains only a minimal annotation, and certainly never constitutes a closure. Take the following cases, for example, taken from NYT19990505.0443.tml in the signal-augmented corpus and edited slightly for brevity:

(5.9) A jogger <EVENT eid="e64">*observed*</EVENT> Kopp's car <SIGNAL sid="s7">*at*</SIGNAL> <TIMEX3 tid="t10">*6a.m.*</TIMEX3> near Slepian's home <TIMEX3 tid="t11">*10 days*</TIMEX3> <SIGNAL sid="s8">*before*</SIGNAL> the <EVENT eid="e65">*murder*</EVENT>, and, <EVENT eid="e66">*curious*</EVENT> why a stranger would be <EVENT eid="e67">*parked*</EVENT> there so early, <EVENT eid="e68">*wrote*</EVENT> down the license plate number.

In this section, our approach found the links listed in Table 5.30 (in this example, event eids and instance eiids have a 1:1 mapping, so ei65 corresponds to event e65).

Signal ID	Argument 1	Argument 2	In GS?
s8	ei64	ei65	yes
s8	ei65	ei66	no
s8	ei65	ei67	no
s8	ei65	ei68	no
s8	ei65	t1	no
s8	ei65	t11	yes

Table 5.30: Sample signals and arguments found in N45

Many of the links suggested but not annotated are in fact correct from the text. For example; signal s8 (*before*) is said to describe the temporal relationship between ei65 *murder* and *curious*, which it does, as well as e.g. ei65 *murder* and ei68 *wrote*, which is also a correct description of that temporal relationship. However, these relations are not in the gold standard annotation (despite being correct interpretations of the text) and so they present as false positives. Because manual examination of all the false positives to detect errors of this kind would be time consuming, the 26.2% figure that comes from automatic evaluation must be seen as a lower bound.

For a more concrete evaluation, one can constrain the set of signal associations considered to that described by TLINKs in the document. That is, we assume that events and timexes are known, and also that interval pairs (as in TLINK arguments) have been identified, and that the remaining tasks in a document's TimeML annotation are signal annotation and then TLINK relation type assignment. To this end, one only considers pairs of intervals that are also found in the gold standard. Thus, the evaluation problem is constrained somewhat, excluding the implicit temporal relation identification stage the initial evaluation includes. Therefore, this is referred to as the "constrained joint approach". It is implemented by, instead of using a window to choose interval pairings for consideration, using the pairing suggested in each of the annotated TLINKs.

In this case, there are 136 gold standard entities again. Result are given in Table 5.31. The system finds 99 signalled interval pairs that have arguments corresponding to a TLINK in the gold standard. Of these 99, 88.9% (88) are correct annotations (e.g. precision is 88.9%); the remaining 11 are spurious. This gives a recall of 64.7% and F1 of 74.9%. We describe these with F1 and not the Matthews correlation coefficient often associated with evaluating binary classifiers because the set of true negatives is very large in this case but not very interesting, and F1 does not take them into account.

In summary, using no signal information from the gold standard and simply relying on models for signal annotation, we achieve a 74.9% F1 rate for the overall joint task of identifying temporal signal expressions and linking each expression found to a pair of intervals that it temporally co-ordinates.

Signal/TLINK associations	Count	Proportion
In N45	136	-
Found	99	-
Found, both args in N45	88	88.9%
Signal in N45, new TLINK assoc	0	0.00%
Found based on new signals	11	11.1%

Table 5.31: Details of the constrained joint approach to signal annotation.

Corpus	TLINKs	Non-signalled	Signalled	Signal %
AQN45	1 048	932	116	11.1%
AQN45-sig	1 062	915	147	13.8%

Table 5.32: TLINK stats over corpora used for extrinsic evaluation

5.8.2 Combined Signal Annotation and Relation Typing

We know that signals are helpful in informing TLINK labelling. We also know that we can automatically annotate signals, to a reasonable degree of accuracy. It remains to be seen whether this degree of accuracy is sufficient for automatically-created signal annotations that are of overall help in TLINK labelling. It may be that the TLINK labelling information provided by signals is offset by imperfect automatic signal annotation, or that false positives in signal annotation provide misleading and counter-productive information to TLINK labelling.

In this section, experiments are reported whose aim is to determine whether automatic signal annotation has an impact on the overall task of TLINK labelling. We take the N45 section of the AQUAINT corpus as the dataset. It is curated to add missing signals, intervals and associations (details in Table 5.32). Two experiments are conducted. The first, a baseline, is over the manually signal-augmented version of the N45 docs (AQN45-sig) using a link labelling model trained on TB-sig, including no signal-specific features. This ignores temporal signals and represents the situation where a gold standard annotation is performed and a model learned without any signal information, and evaluated over unseen data. The second experiment uses TB-sig to learn models for signalled and non-signalled TLINKs, using the signal features described in Section 5.3.1, and then evaluates the performance of these models at labelling their respective parts of the automatically signal annotated version of N45 described in Section 5.8.1. This represents the scenario of having already annotated events, timexes and pairing intervals, then doing automatic signal annotation on unseen data, and evaluates how helpful these signal annotations are for TLINK labelling. We exclude new TLINKs identified in the course of automatic signal association, as we have no gold standard the relation type of these. The version of N45 with automatically generated signal annotations is referred to as AQN45-auto.

The distribution of interval pair types and TLINKs in the training data, TB-sig, is shown in Table 5.33. Similar data for evaluation corpora is in Table 5.32.

Interval types	Non-signalled	Signalled
Event-event	3 179	343
Event-time	2 299	529
Time-time	126	14

Table 5.33: Training dataset sizes from TB-sig used for signal annotation models

Corpus	Subset of links	Event-Event	Event-Time	Overall	Baseline
AQUAINT N45 plain	all	44.0%	56.4%	55.8%	28.9%
AQN45-auto	all	62.0%	58.4%	58.6%	28.9%
AQN45-auto	unsignalled	50.0%	58.6%	58.5%	28.4%
AQN45-auto	only signalled	66.7%	56.8%	59.2%	32.0%
AQN45-sig	only signalled	70.5%	72.2%	71.64%	32.8%

Table 5.34: TLINK labelling accuracy over corpora used for extrinsic evaluation. The baseline is the overall most-common-class for TLINKs in the training data (TB-sig). Interval text features are not included. There were no timex-timex links. The difference between the first two rows shows the impact that this total asignal discrimination and association approach has on TLINK labelling accuracy.

It can be seen that TLINKing based on automatic signal annotations, detailed in the second row (AQN45-auto / all) of Table 5.34, performs better than TLINKing with no signal information (the first row). The approach is therefore effective.

However, signalled TLINKs in the gold standard are still labelled substantially better than when automatic signal annotations are used (compare the fourth and fifth rows). Event-event links tend to draw particular benefit from signal annotations (see second and third columns), and this is still the case with automatic signal annotations; 66.7% accuracy was achieved on the signalled event-event links, and 70.5% using gold-standard links, compared to only 44.0% labelling accuracy without any signal information. Overall, event-event temporal relation typing performance on this dataset increased from 44.0% accuracy ignoring signals to 62.0% when using automatically annotated signals – an 18.0% performance increase, or 32.1% error reduction.

The N45 part of the AQUAINT corpus unfortunately has a much lower event-event : event-timex TLINK ratio than TimeBank, with only 50 event-event vs. 1 012 event-time links (4.71% of the whole). For comparison, TB-sig has 2 828 event-time links to 3 522 event-event; event-event comprise 55.5% of links. The bias in N45 has therefore led to an underestimate of the extra impact that signal information has on general event-event labelling. Nonetheless, the results confirm the efficiacy of the automatic signal extraction method, and show an overall 2.8% absolute improvement in TLINK labelling over data without signals.

5.9 Chapter Summary

Temporal signals are an important source of information for temporal relations.

This chapter presented a principled investigation into temporal signals and the role they play

in relating and ordering events and times within discourse.

It first presented a linguistic account for temporal signals, followed by a demonstration of their utility in the relation typing task, with a prototype supervised learning approach to temporal relation typing with signals that achieved error reduction of 53% compared to the same system without signal information.

Given this strong motivation for exploring signals, a corpus analysis of temporal signals was conducted, examining an existing TimeML-annotated corpus. This was followed by a brief attempt at automatic temporal signal annotation which quickly revealed insufficient quality in signal annotations. As a result, the corpus was re-annotated with extra signals, including the events, timexes and temporal relations that the new signals required. This resource is made publicly available, as TB-sig.

Having a strong corpus, an approach for automatic signal annotation could be developed. This was taken as a two-part task. Firstly, as many signal expressions are polysemous, one must determine which occurrences of candidate signal words occur having a temporal sense. This was achieved with 83.0% precision. Secondly, given a signal, one must determine which temporal intervals it co-ordinates. Two approaches to this problem were addressed – one considering intervals one at a time and ranking them, then assuming that the top two are linked, and another considering each possible pair of intervals. The interval pair approach worked best, achieving 83.5% precision.

Having developed both stages of the signal annotation mechanism, these were evaluated jointly against a new gold-standard signal corpus derived from the AQUAINT TimeML corpus. With the least-constrained, hardest evaluation technique, 64.7% of the gold-standard annotations were found automatically by the discrimination/association system proposed in this chapter.

Finally, with a full signal annotation system developed, the impact of automatic signal annotation on the overall task of temporal relation typing was evaluated. Results were positive. Adding automatic signal annotations and then feature representations of these automatically-found signals improved the absolute performance of a temporal relation type classifier by 18% for event-event links and 2.0% for event-time links.

In summary, we showed that temporal signals were useful in temporal relation typing, and developed approached for automatically annotating them, which performed well enough to give a net performance increase in the temporal relation typing task.

Chapter 6

Using a Framework of Tense and Aspect

> For years I have endeavored to break through the veil which shrouded it, and at last the time came when I seized my thread and followed it.
>
> *The Final Problem*
> Sir Arthur Conan Doyle

6.1 Introduction

Analysis of the temporal relation typing problem in Chapter 4 suggested two directions for investigation. In addition to signals, tense and aspectual differences are prevalent in difficult links. In an attempt to improve performance at typing of temporal relations, this chapter investigates a linguistic framework for tense and aspect.

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

Event-event relations are the hardest to label (Chapter 4). Around 45% of links in TempEval (a temporal annotation evaluation exercise, see Section 3.4.4) event-event tasks cannot reliably be labelled automatically (see Section 4.2.2). Further, verb-verb links make up a significant amount of the difficult links identified in Section 4.2.

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

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

an event and timex in the same sentence, where the timex is a date or time.

Data-driven approaches to the relation typing task are hampered in two ways. Firstly, there is a shortage of ground truth training data, which is in turn partially due to the high cost of annotation. As Lapata and Lascarides (2006) point out, this leads to low volumes of instances for many combinations of tense and aspect values for pairs of events (see Table 6.1), potentially hampering automatic hypothesis learning. Secondly, the variation of expression annotatable using TimeML is relatively limited, describing three "tenses"[1] (past and past participle, present and present participle, and future) and three "aspects" (none, perfective and progressive). This markup language may be insufficiently descriptive to capture the relations implied by all the variations in linguistic use of tense and aspect.

Reichenbach (1947) offers a theoretical framework for analysis of tense and aspect that can be used to predict constraints on temporal orderings between verb events based on their tense and aspect, and also between times and tensed verbs. Applying Reichenbach's framework requires tense and aspect information, which is provided in TimeML (meaning that it might be possible to apply this framework without a major annotation effort).

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

It may be the case that machine learning methods are unable to make effective use of the tense information available in TimeBank. Phenomena such as tense shifts between events have been shown to help humans temporal ordering (Harris and Brewer, 1973), and therefore may convey some temporal information. However, the percentage of links with tense shifts is roughly the same in the general case (40% in TimeBank) and the difficult link set (36%). As these figures are roughly the same, it may be that supervised approaches fail to make generalisations that take advantage of the information given in tense shifts.

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

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

The rest of this chapter is structured as follows. Firstly, we discuss in abstract terms a conceptual model for time. Second, there is an introduction to Reichenbach's framework and a description of how it interacts with temporal expressions as well as verb events, followed by a summary of related work. Next, validation of the framework is attempted by describing how the framework can

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

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

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

be related to TimeML and then an evaluation of it against ground truth temporal relation type information. The framework's relation type constraints are then applied to the temporal relation typing task alongside data from TimeML annotations, as part of a machine learning approach to relation typing, and results presented. It is found that Reichenbach framework is potentially helpful. To allow inclusion of what the framework provides that is not in TimeML already, an annotation scheme for the framework is introduced (RTMML) which may also be used as an extension to TimeML. Finally, the chapter concludes with a discussion of applications of the framework and future work.

6.2 Timelines in Language

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

Time is often described using the same language as space, as touched upon in Section 5.4.4. We talk about *time travel*, use words such as *faster*, *before* and *at* and specify directions such as *forward* and *backward*. The linguistic relation between expression of time and space is sometimes taken to extremes; some have suggested that we travel through time facing backwards, because we can only see the past and not the future (Pratchett, 1986). The spatio/temporal polysemy is even learned by classifier models when attempting to detect temporal usages of words (Section 5.6.4). This linguistic similarity is rooted in the way that humans understand non-literal motion (such as

in temporal transitions) using the same cognitive resources as we understand literal (e.g. spatial) motion (Matlock et al., 2005).

Given that time is a linear and effectively continuous (Huggett, 2010) dimension which progresses unidirectionally (Eddington, 1928) but can be conceived of in either direction (Stocker, 2012), we talk about its description in language with a model of time as uni-dimensional (cf. McTaggart's A-series (McTaggart, 1908)).

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

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

These three parts correspond directly to the rudiments of tense in language; the past tense, present tense and future tense permit expression of events within the past, at the present, or within the future part of a timeline (cf. McTaggart's B-series). Thus, simple tense usage allows positioning of events within regions on a timeline relative to the present; and so, in that it describes temporally relative points, tense is inherently deictic (Lyons, 1977; Michaelis, 2006). The tenses corresponding to these three categories are known as **absolute tenses**.

Given such a tense structure, one may identify two temporal points upon the timeline. One is the time at which the description of the event is uttered or perceived, and the other, that may be in any of the three timeline parts, corresponds to the time that the described action took place. This simple structure allows us to temporally express events relative to the present.

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

In language, this double-deixis can be accounted for in a system of tense and aspect. It is

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

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

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

6.3 Description of the framework

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

6.3.1 Time points

To describe a tense, Reichenbach introduces three abstract time points. Firstly, there is the speech time,[3] S. This represents the point at which the verb is uttered or written. Secondly, event time E is the time that the event introduced by the verb occurs. Thirdly, there is reference time R; this is an abstract point, from which events are viewed. Klein (1994) describes it as "the time to which a claim is constrained".

In Example 6.1, speech time S is when the author created the discourse (or perhaps when the reader interpreted it).

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

Reference time R is *then* – an abstract point, before speech time, but after the event time E, which is the leaving of the building. In this sentence, one views events from a point in time later than they occurred. Therefore, the final configuration is $E < R < S$.

[3]For this thesis, speech time is equivalent to DCT, unless otherwise explicitly positioned by discourse. Under (Fillmore, 1971)'s description, this is the same as always setting speech time S equal to encoding time ET and not decoding time DT.

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

Table 6.2: Reichenbach's tenses; from Mani et al. (2005)

6.3.2 Reichenbachian Tenses

Reichenbach details nine tenses (see Table 6.2). The tenses detailed by Reichenbach are past, present or future, and may take a simple, anterior or posterior form. In English, these apply to single non-infinitive verbs and to verbal groups consisting of head verb and auxiliaries. The tense system describes abstract time points for each tensed verb and how they may interact, both for a single verb and with other events described by verbs.

In Reichenbach's view, different tenses specify different relations between E, R and S. Table 6.2 shows the six tenses conventionally distinguished in English. As there are more than six possible ordering arrangements of S, E and R, some English tenses might suggest more than one arrangement. Reichenbach's named tenses names also suffer from this ambiguity when converted to $S/E/R$ structures, albeit to a lesser degree. When following Reichenbach's tense names, it is the case that for past tenses, R always occurs before S; in the future, R is always after S; and in the present, S and R are simultaneous. Further, "anterior" suggests E before R, "simple" that R and E are simultaneous, and "posterior" that E is after R. The flexibility of this framework is sufficient to allow it to account for a very wide set of tenses, including all those described by Song and Cohen (1988), and this is sufficient to account for the observed tenses in many languages. Past, present and future tenses imply $R < S$, $R = S$ and $S < R$ respectively. Anterior, simple and posterior tenses imply $E < R$, $E = R$ and $R < E$ respectively.

6.3.3 Verb interactions

While each tensed verb involves a speech, event and reference time, multiple verbs may share one or more of these points. For example, all narrative in a news article usually has the same speech time (that of document creation). Further, two events linked by a temporal conjunction (e.g. *after* - see Chapter 5) are very likely to share the same reference time. Basic methods of linking between verb events or linking verbs to fixed points on a time scale are described below.

Special properties of the reference point

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

Permanence Firstly, when sentences are combined to form a compound sentence, tensed mean verbs interact, and implicit grammatical rules require tenses to be adjusted. These rules operate in such a way that the reference point is the same in all cases in the sequence. Reichenbach names this principle **permanence of the reference point**; *"We can interpret these rules as the principle that, although the events referred to in the clauses may occupy different time points, the reference point should be the same for all clauses"*. Figure 6.1 contains an example of this principle.

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

In Example 6.2, an explicit time (*10 o'clock*) determines our reference point through positional use.

(6.2) *It was 10 o'clock, and Sarah had brushed her teeth.*

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

Example Reichenbachian verb-verb links

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

(6.3) *In February 1917, the Germans landed their offensive. By April 26th, it was all over.*

Example 6.3 shows a temporal expression describing a day – April 26th. The expression is ambiguous because we cannot position it absolutely without knowing which year it refers to. This type of temporal expression is interpreted with respect to reference time, not with respect to speech

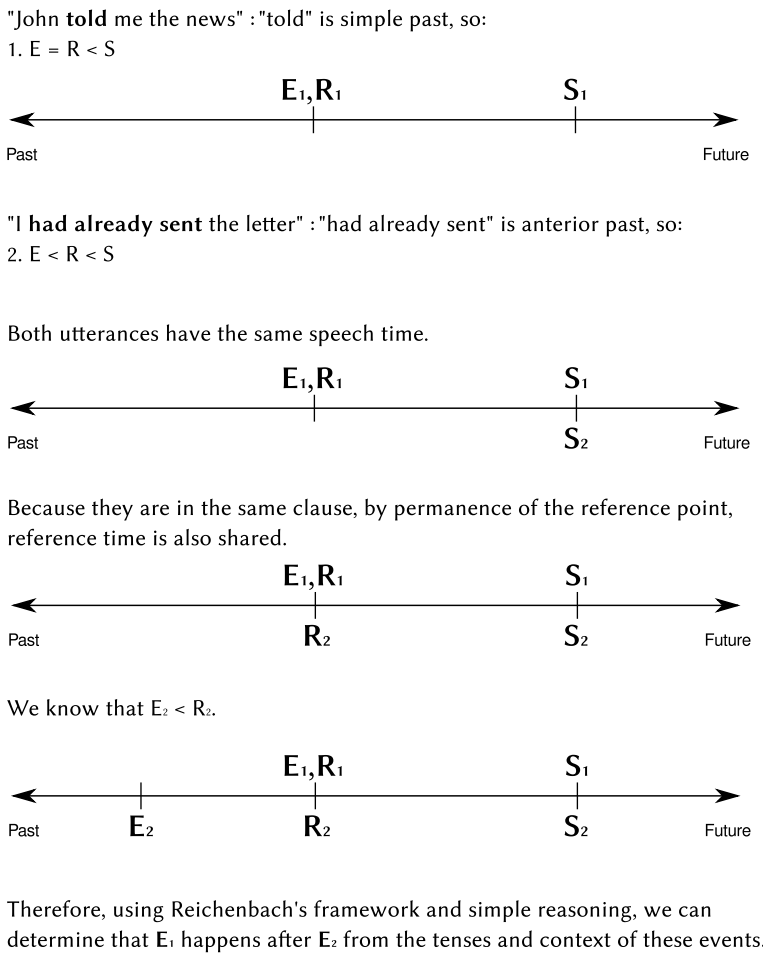

Figure 6.1: An example of permanence of the reference point.

time (Ahn et al., 2005). Without a time frame for the sentence (presumably provided earlier in the discourse), we cannot determine which year the date is in. If we are able to set bounds for R in this case, the time in Example 6.3 will be the April 26^{th} adjacent to or contained in R; as the word *by* is used, we know that the time is the April 26^{th} following R, and can normalise the temporal expression, associating it with a time on an absolute scale.

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

Example 6.4 and Figure 6.1 show a sentence with two verb events – *told* and *had sent*. Using Reichenbach's framework, these share their speech time S (the time of the sentence's creation) and reference time R, but have different event times. In the first verb, reference and event time have the same position. In the second, viewed from when John told the news, the letter sending had already happened – that is, event time is before reference time. As reference time R is the same throughout the sentence, we know that the letter was sent before John mentioned the news. Describing S, E and R for verbs in a discourse and linking these points with each other (and

with times) is the only way to ensure correct normalisation of all anaphoric and deictic temporal expressions, as well as enabling high-accuracy labelling of some temporal links.

Example 6.5 contains a more advanced example. It shows a pair of temporally related verbs taken from the list of difficult links found earlier (see Section 4.3.1).

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

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

6.3.4 Temporal context

In the linear order that events and times are introduced in discourse, speech and reference points persist until changed by a new event or time. Observations during the course of this work suggest that the reference time from one sentence will roll over to the next sentence, until it is repositioned explicitly by a tensed verb or time. To make discussion of sets of verbs with common reference times easy, we call each of these groups a **temporal context**.

To cater for subordinate clauses in cases such as reported speech, we add a caveat – S and R persist as a discourse is read in textual order, for each temporal context. A context is an environment in which events occur, and may be the main body of the document, a tract of reported speech, or the conditional world of an *if* clause (Hornstein, 1990). For example:

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

Here, *said* and *changed* share speech and reference points. Emmanuel's statement occurs in a separate context, which the opening quote instantiates, ended by the closing quote (unless we continue his reported speech later), and begins with an S that occurs at the same time as *said* – or, to be precise, *said*'s event time E_{said}.

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

As described in Chapter 4 of Hornstein (1990) in his description of the sequence of tenses with regard to Reichenbach's framework, permanence of the reference point does not apply between main events and embedded phrases, relative clauses or quoted speech. These occur within a separate temporal context, and it is likely that they will have their own reference time (and possibly even speech time, for example, in the case of quoted speech). In order to apply permanence of the reference point, it ought only be applied within the same temporal context. Verbs to which

permanence may be applied are said by Reichenbach to be those to which the grammatical rules of the **sequence of tenses** (an abstract set of grammatical rules not described in his paper) apply. Different contexts will have a consistent reference point, and so permanence of the reference point may be applied to verbs within that context in order to gain information about their temporal relations. Permanence does not apply across different temporal contexts.

Dowty (1986) hints at the concept of temporal context with the idea of the **temporal discourse interpretation principle** (TDIP). This states:

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

The TDIP accounts for a set of sentences which share a reference and speech point. However, as with other definitions of temporal context, this principle involves components that are difficult to automatically determine (e.g. "consistent with definite time adverbials").

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

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

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

6.3.5 Quoted Speech

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

Example 6.7 shows two verb events: one initiates quoted speech (*told*), and the other is within this reported speech (*hold*).

(6.7) *This morning General Powell told reporters, "We will hold a press conference shortly."*

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

Exposition of the principle benefits from Hornstein (1990)'s modestly extended definition of speech time, as follows:

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

6.3.6 Limitations of the framework

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

Limited tenses

The included tenses and aspects are insufficiently expressive to cover the gamut of linguistic expressions of temporality. One may look at lexical semantic models of tense and aspect in English to discover a wider inventory of possible tenses and aspects in that language (By, 2002), or examine other languages with richer aspect systems to see what the framework glosses over in those cases (e.g. Paslawska and van Stechow (2003)). Limitations of the Reichenbachian perfect can be seen from Table 6.2, where there is more than one triple that corresponds to the future perfect. Nevertheless, many tense and aspect systems can be described in terms of Reichenbach's framework, albeit not always as a 1:1 mapping.

Progressive aspect

The progressive is used for events that have both a start and end and are currently ongoing; that is, in-progress activities. This makes it possible to refer to points within an event. However, Reichenbach's framework is point-based, and point-based temporal algebras generally assume that when point events are referenced, they are only referenced in terms of being before, after or simultaneous with another temporal entity. This makes it difficult to accurately represent more complex verbal event structures. Introducing interval reasoning to the framework can help (that is, dealing with intervals in terms of start and end points, instead of a single point for the whole), although it is sufficient to achieve this through treating events as a coupled start and end point

(where the start is never after the end). This has the advantage of permitting semi-interval type reasoning (see Section 3.2). We discuss this further in Section 6.4.2

on dates

Positional use of the reference point tells us that R is equivalent to a timex in the clause, if given. Because the algebra the framework uses to describe tenses is point-based, the start and end of the given time period are equal to the start and end of the reference time. This gives problems when a described event takes place during a provided timex, but does not have the same start and stop times. Example 6.8 is taken from Hinrichs (1986):

(6.8) *Mary left England on May the 22nd, 1979*

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

Non-English tense system

Some languages are difficult to accommodate in Reichenbach's framework. To accommodate Russian, for example, one must make specific and extensive additions to the framework, including binary temporal relations between points for each verb (Giorgi and Pianesi, 1997). Such a system can be extended to cover a large range of Slavic languages (Hristova, 2006), though is too complex to implement for a first attempt at automated temporal annotation using Reichenbach's framework.

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

(6.9) 我 吃 吗 (wǒ chī mǎ) – "I eat horse"

A simple sentence is given in Example 6.9. This can be interpreted in English as *"I prefer to eat horse"*, *"I am currently eating horse"* or *"I will eat horse"*, *"I ate horse"*; contextual markets are required for clarification. The default interpretation is that of simple present tense. Past tense can be signified with guō (过), and completion with le (了), both of are placed directly after the verb. It is therefore possible to capture the relation between speech and event points, and we can determine if the reference point is after the event or not. There is nothing to clarify the difference between simple and anterior tenses, and (as in English) the simple present is also

used to indicate habitual truths (e.g. *I eat horse*). However, unlike English, the simple present progressive (e.g. *I am eating horse*) looks identical to the habitual use. Further information is expressed through temporal adverbials and not considered tense. The general lack of inflection or cohesive verb groups suggests that Reichenbach's framework can only be applied to Chinese in a limited fashion, decreasing its general utility.

Split reference point

Some tensed temporal descriptions of events are difficult to framework with just a single reference point. For example, from Prior (1967):

(6.10)
- *"I shall have been going to see John."* (that is, there is some point in the past at which I anticipated seeing John; note this is not a description of habitual behaviour)
- $S < R_1 < E < R_2$

It is true that the tenses and abstract points provided by the three-point framework are insufficient to capture this statement, without invoking an extra verb event. However, in TimeBank no such contrived utterances were found during candid examinations or error analysis from applying the framework to predict TimeML relations.

Reification of the reference point

Tanaka (1990) takes exception to the abstract nature of the reference point, and that it is never reified or explicitly lexicalised. He questions the requirement for reference time in a system of tense, and raises a few examples that are difficult to express using Reichenbach's framework. Tanaka's criticism and example are as follows.

(6.11)
- Now Megumi will marry Kazuhiko next month.
- $S < E = R$

In Example 6.11, the temporal adverbial *next month* is used to position the reference point, R. With the tense used here – simple future – this also places E (the time of marrying) during *next month*, which is the correct interpretation. However, Tanaka suggests that the framework does not explain the influence of *Now* in this sentence; for which verbs does it fix the reference point? This criticism could be viewed as a variation on the requirement for two reference points to describe some verbs.

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

6.4 Validating the Framework Against TimeBank

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

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

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

Secondly, Reichenbach is vague about temporal context. It is unclear from TimeML annotations alone which sets of verbs can be considered to be in the same "temporal context" (see Section 6.3.4). Reichenbach simply states that the framework is intended to follow the sequence of verbs. The descriptions of the "sequence of tenses" suggest it is difficult to implement programatically with current technology (see e.g. Chapter 4 of (Hornstein, 1990)), and require accurate identification of reported speech, embedded phrases, relative clauses, reference-time shifting temporal adverbials and so on. This presents a number of complex syntactic and linguistic scoping tasks that may be difficult to perform automatically. Therefore, one needs an approximation of temporal context in order to choose which verb pairs to attempt to relate.

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

In the two-event sentence of Example 6.12, *fished* is anterior present with arrangement $E <$

$S_1 = R_1$ and *eat* is simple future, with arrangement $S < R_2 = E_2$.

(6.12) *"I have fished$_1$; John will eat$_2$."*

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

Note that eat_2 could be interpreted as Reichenbachian posterior present, with arrangement $S = R_2 < E_2$. This gives the same temporal ordering of events, but through transitivity permits a shared reference point (i.e. $R_1 = R_2$). In this situation, as is sometimes the case in English, it is not possible to decide precisely which of posterior present and simple future applies. However, this is of little impact in this toy example when we are concerned primarily with determining relations between events; the reference point is only a means to that end.

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

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

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

6.4.1 Minimal Interpretation of Reichenbach's Framework

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

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

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

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

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

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

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

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

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

As can be seen from prevalence of `vague` entries in the table, many combinations of tense offer no helpful constraint in terms of Allen's interval temporal relations. This is a hint that this

e1 ↓; e2 →	**Sim Past**	**Pos Past**	**Ant Pres**	**Sim Pres**	**Ant Fut**	**Sim Fut**
Sim Past	vague	after	vague	after	after	after
Pos Past	before	vague	vague	vague	after	after
Ant Pres	vague	vague	vague	after	vague	after
Sim Pres	before	vague	vague	overlap	vague	after
Ant Fut	before	before	vague	vague	vague	after
Sim Fut	before	before	before	before	before	vague

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

Output	**Count**	**Consistent**	**% consistent**
after	14	4	28.6%
overlap	19	15	84.2%
before	45	12	26.7%
Total	78	31	39.7%
vague	68	-	-

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

particular interpretation of Reichenbach's tense may not see great performance increases when used for relation typing, and (depending on the actual distribution of tenses in the corpus) may not give a very clear picture of how accurate Reichenbach's model is.

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

Minimal Interpretation Failure Analysis

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

No progressive The framework doesn't handle the progressive aspect. If events have differing tenses (e.g. present and then future), the framework suggests by means of transitivity that the event time of the present-tensed verb is before that of the future-tensed verb. This makes this implicit assumption that the present-tensed item will have completed before the future-tensed item begins, ruling out any possibility of overlap. Progressive aspect is used as an indicator of ongoing processes, and could be used to weaken the constraint imposed by this minimal interpretation. For example, in *"I am running. Heston will cook."*, it is not certain that I will have finished running

before the point that Heston starts cooking; that is to say, overlap is possible.

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

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

6.4.2 Advanced Interpretation of Reichenbach's Framework

The interpretation of Reichenbach's framework described above makes a few simplifications, and the results are poor. These simplifications may be the cause of incongruence between the framework's apparent suggestions and human-annotated ground-truth data. We improve the interpretation of Reichenbach's framework in the following ways, and re-check it. Some of this section's material also appears in Derczynski and Gaizauskas (2013a).

Account of progressive aspect: In TimeML, aspect values are composed of two "flags", `perfective` and `progressive`, which may both be asserted on any tensed verb. Which Reichenbach's basic framework provides an account of the perfect (which TimeML calls perfective), it does not do the same for the progressive. This is resolved by splitting the event time E into start and finish points E_s and E_f between which the event obtains, as also done by e.g. Kowalski and Sergot (1989). For the simple tenses (where $R = E$), described as having TimeML aspect of NONE, it is assumed not that the event is a point, but that the event is an interval (just as in the progressive) and the reference time is *also* an interval, starting and finishing at the same times as the event (e.g. $R_s = E_s$ and $R_f = E_f$).

Variations of context assignment: Reichenbach's definition of which verbs may be linked through permanence of the reference point is a little vague, described as those that share a common reference point. This is approximated in a number of ways, results of each of which are presented:

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

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

by considering all verb events in the same sentence; by considering all verb events in the same or an adjacent sentence; and by considering all verb events that have a common arrangement of both speech and reference time (e.g. all have the same arrangement of S and R). Ideally one should like to be able to track the speech and reference point through discourse, accounting for relative clauses, embedded phrases, reported speech and the like; in absence of a concerted investigation into performing these tasks reliably automatically, these approaches are approximations.

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

How to interpret relations suggested by the framework: Previously a label from one of four classes (before, after, overlap, vague) was assigned to a temporal relation, based on the tenses of its participant verb events. These classes did not accurately capture the 14 TimeML relations, and in many cases represented a disjunction of possible interval relation types. Working on the hypothesis that Reichenbach's framework may constrain a TimeML relation type to more than just four possible groupings, the table of tense-tense interactions is rebuilt, giving for each event pair a disjunction of TimeML relations instead of one of four labels. This has the advantage of adding distinctions that the minimal framework could not capture. Example 6.13 and Example 6.14 would both be labeled "before" under that scheme, even though the latter is ambiguous regarding whether the progressive event has finished, and could signify an overlap.

(6.13) *Anne had eaten breakfast. Bernard will sing.*

(6.14) *Chris was cleaning windows. Diana will sleep.*

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

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

In this case, Example 6.13 suggests the TimeML relation *eaten* BEFORE *sing*, whereas because the end point of *cleaning* is not certain in Example 6.14, any of BEFORE, INCLUDES, or ENDED_BY may apply between *cleaning* and *sleep*. In this way, and with other arrangements of the speech, event and reference time, resolving relation types to disjunctions of potential interval relations provides a richer, more descriptive and more precise way of capturing the framework's output. An example is given in Table 6.7.

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

This finite set of interval relation disjunctions overlaps with the relation types grouped by Freksa (Section 3.2). For example, for two events E_1 and E_2, if the tense arrangement suggests that E_1 starts before E_2 (for example, E_1 is simple past and E_2 simple future), the available relation types for E_1/E_2 are BEFORE, IBEFORE, DURING, ENDED_BY and INCLUDES.

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

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

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

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

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

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

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

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

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

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

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

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

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

In general, consistency is better than with the minimal interpretation discussed above. The "same SR" context gives good results, though has limited applicability in that it considers comparatively reduced sets of TLINKs (e.g. only half of same-sentence links). As both arguments having the same S and R occurs when they have the same TimeML tense, the only variant in these cases – in terms of data that contributes to Reichenbachian interpretation – is the TimeML aspect value. The increased "coverage" of the framework when given the constraint that TLINKs in which both arguments have the same TimeML tense hints that this is a critical factor in interpreting tense, and considering it may lead to improvements in temporal relation typing techniques that rely on aspect, such as that of Costa and Branco (2012). The overall result is that Reichenbach's framework is capable of suggesting helpful relation types in some situations, and suggests further effort in applying and using the framework.

A slightly extended, standalone version of this validation can be found in (Derczynski and Gaizauskas, 2013a).

6.5 Applying Reichenbach's Framework to Temporal Relation Typing

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

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

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

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

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

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

TimeBank are used, as well as the TLINKs identified there; the only task addressed is that of temporal relation typing.

6.5.1 Same context event-event links

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

This situation applies to any two verb events that have a shared reference point. Verb events are identifiable by the event having a TimeML POS attribute of VERB, excluding those with a `tense` of `NONE` or `INFINITIVE`. A shared reference point is assumed for all verbs in the same sentence. Sentences are split using the Punkt sentence tokeniser for English (Kiss and Strunk, 2006). These experiments use the minimal interpretation of Reichenbach's framework, described above.

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

Results

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

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

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

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

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

Further work would concentrate on better discriminating which cases can be considered for

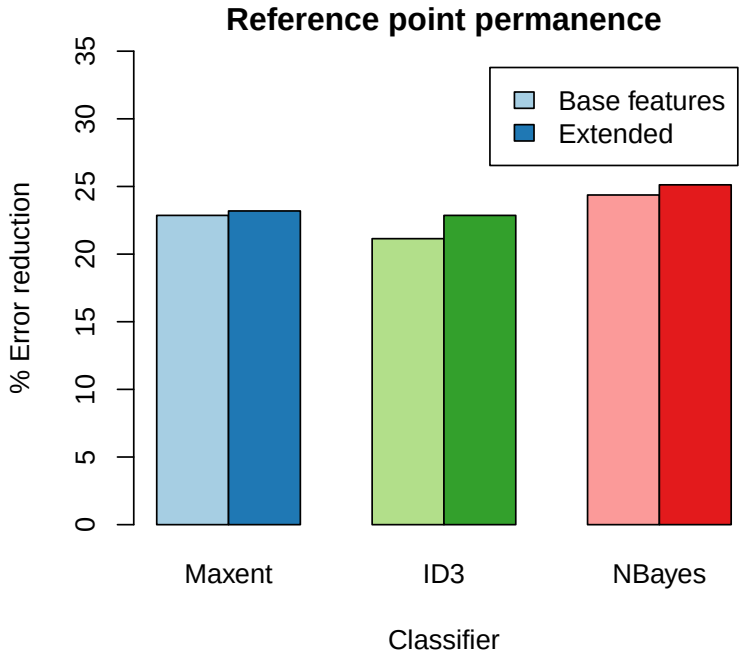

Figure 6.2: Error reduction in SCEE links with and without features representing permanence of the reference point, modelling temporal context as same-sentence. The darker coloured columns correspond to error reduction using the feature derived from advanced interpretation of Reichenbach's framework.

application of permanence of the reference point. These are likely to span sentences. An annotation for delimiting these cases (e.g. temporal contexts) is put forward later, in Section 6.6.

6.5.2 Same context event-timex links

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

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

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

Features

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

Dependency parses (generated by the Stanford Parser (De Marneffe et al., 2006)) help determine whether or not a timex and event are syntactically connected. These parses also yield some extra information, which is included as features. These are:

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

Results

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

Results are given in Table 6.13. The extended features offered a performance improvement from 20.18% error reduction to 24.26% error reduction for the best-performing classifier (maxent). Performance with just the Reichenbach E/R determining feature are also included in the table.

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

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

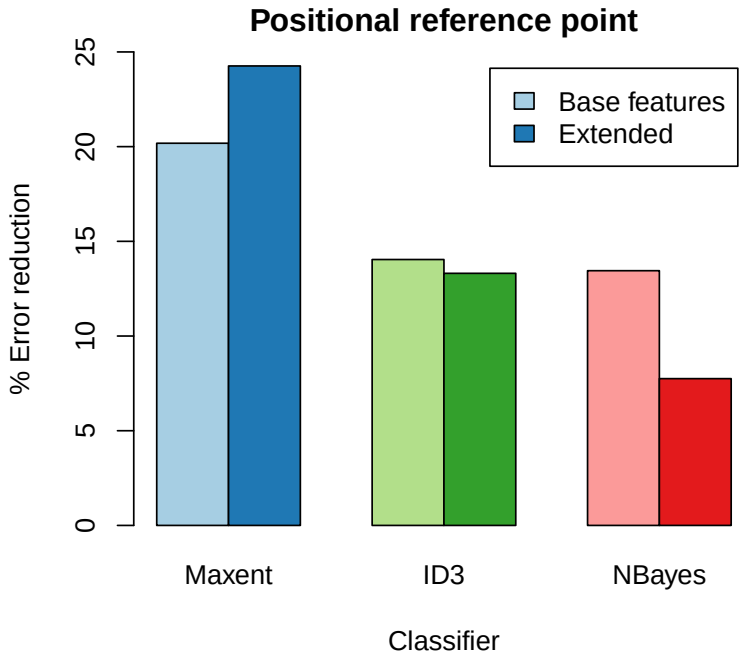

Figure 6.3: Comparative performance on labelling event-time links where the time positions the reference point.

The feature is not as useful on its own as it is with the three other dependency-graph derived features.

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

6.5.3 Summary

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

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

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

6.6 Annotating Reichenbach's Framework

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

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

The annotation schema RTMML is intended to describe the verbal event structure detailed in Reichenbach (1947), in order to permit the relative temporal positioning of reference, event, and speech times. A simple approach is to define a markup that only describes the information that we are interested in, and can be integrated with TimeML. For expositional clarity we use

our own tags but it is possible (with minor modifications) to integrate them with TimeML as an extension to that standard.

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

6.6.1 Motivation for annotating the framework's points

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

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

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

TimeML's `aspect` attribute will inform us if the reference time is after the event time; that is, if the event is "complete" (to gloss over linguistic nuances detailed by Vendler (1957)) before the time of reference point. This distinguishes two classes; TimeML `aspect:PERFECTIVE` corresponds to $E < R$, and `aspect:NONE` corresponds to $E \not< R$ (that is, a conflation of $E = R$ and $R < E$).

Also, TimeML does not address the issue of annotating Reichenbach's tense framework with the goal of understanding reference time or creating resources that enable detailed examination of the links between verbal events in discourse. It is not possible to describe or build relations to reference points at all in TimeML.

6.6.2 Proposed solution

Here we discuss what should be annotated in order to capture the information described by Reichenbach's framework, and put forward an annotation schema. Some of this section's material overlaps with Derczynski and Gaizauskas (2011b).

Requirements

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

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

Annotation schema

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

RTMML documents use standoff annotation. This keeps the text uncluttered, in the spirit of *ISO LAF*[4] and *ISO SemAF-Time*.[5] Annotations reference tokens by their position in the source. Token indices begin from zero. We explicitly state the segmentation plan with the `<seg>` element, as described in Lee and Romary (2010) and *ISO DIS 24614-1 WordSeg-1*.

The general speech time of a document is defined in the `<doc>` element, which has one optional attribute, `@time` (the `@` indicating that `time` is an attribute name). This is either the string `now` or a normalised value, formatted according to TIMEX3 (Boguraev and Ando, 2005) or TIDES (Ferro et al., 2005).

Each `<verb>` element describes a tensed verb group – that is, a sequence of main and auxiliary verbs that comprise a single verb event. The `@target` attribute describes the verb or group's extents, using segment offsets. It has the form `target="#token0"` or `target= "#range(#token7, #token10)"` for a 4-token sequence. Comma-separated lists of offsets are valid, for situations where verb groups are non-contiguous. Every `verb` has a unique value in its `@id` attribute. The Reichenbachian tense structure of a verb group is described using the attributes `@view` (with values *simple*, *anterior* or *posterior*) and `@tense` (*past*, *present* or *future*).

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

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

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

[4]ISO 24612:2012 Language resource management – Linguistic annotation framework (LAF)
[5]ISO 24617-1:2012

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

Table 6.14: RTMML relation types

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

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

6.6.3 Special RTMLINKs

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

To simplify the annotation task, RTMML permits an alternative annotation with the `<rtmlink>` element. The `<rtmlink>` annotation can be used to describe verbs affected by permanence of the reference point (e.g. to reify temporal contexts), positional use of the reference point and positional use of the speech point. This element takes as arguments a relation and a set of times and/or verbs. Possible relation types are POSITIONS, SAME_TIMEFRAME (annotating permanence of the reference point) and REPORTS for reported speech; the meanings of these are given in Table 6.14.

When more than two entities are listed as `rtmlink` targets, the relation is taken as being between an optional `source` entity and each of the `target` entities. Moving inter-verbal links to the `<rtmlink>` element helps fulfil *TEI p5* and the *LAF* requirements that referencing and content structures are separated.

6.6.4 Example RTMML

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

In Example 6.15, we define a time *Yesterday* as `t1` and a verbal event *ate* as `v1`.

(6.15)
```
<rtmml>
Yesterday, John ate well.
 <seg type="token" />
 <doc time="now" />
 <timerefx xml:id="t1" target="
       #token0" />
 <verb xml:id="v1" target="#token3"
```

```
        view="simple" tense="past"
        sr=">" er="=" se=">"
        r="t1" s="doc" />
    </rtmml>
```

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

This sentence includes a positional use of the reference point, that is, where a time-referring expression determines reference time. This is annotated in `v1` when we say `r="t1"` to verbosely capture a use of the reference point. Further, as the default S/E/R structure of a Reichenbachian simple past tensed verb is non-ambiguous, the attributes signfiying relations between time points may be omitted. To simplify the RTMML in Example 6.15, we could replace the `<verb>` element with that in Example 6.16:

(6.16)
```
<verb xml:id="v1" target="#token3"
      view="simple" tense="past"
      s="doc" />
<rtmlink xml:id="l1" type="POSITIONS">
  <link source="#t1" />
  <link target="#v1" />
</rtmlink>
```

Longer examples can be found in the appendices, including an excerpt of David Copperfield in Example C.1.

Comments on annotation

As can be seen in Table 6.2, there is not a one-to-one mapping from English tenses to the nine specified by Reichenbach. In some annotation cases, it is possible to see from a specific example how to resolve such an ambiguity. In other cases, even if view and tense are not clearly determinable, it is possible to define relations between S, E and R. For example, for arrangements corresponding to the simple future, $S < E$. In cases where ambiguities cannot be resolved, one may annotate a disjunction of possible relation types; continuing the simple future example, we could say "$S < R$ or $S = R$" with `sr="<="`.

Some parts of the annotation task present difficulties. During a trial annotation, while annotators could determine the scoping exercise that is temporal context annotation without too much difficulty, directly mapping a verb group to a single Reichenbachian tense schema was hard, and at best tiring. Decomposing this task into pairwise judgements between S, E and R made annotation easier, though when one could often not see all the information required in order to make the correct judgement; as a result, many pairwise annotations were changed after annotators considered distinct but related pairs. Posing the annotation task as one of temporal constraint,

using more concrete ideas (e.g. "From the text, does this event of *John running* obtain at 9p.m.?" instead of "Is T_9 during E_7?") may reduce annotator fatigue and error. RTMML does not address intentionality, leaving this to annotators and, where expressable, TimeML (which includes the I_ACTION and I_STATE event classes for this purpose).

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

Integration with TimeML

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

6.7 Chapter Summary

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

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

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

Chapter 7

Conclusion

> If we have learned one thing from the history of invention and discovery, it is that, in the long run – and often in the short one – the most daring prophecies seem laughably conservative.
>
> *The Exploration of Space*
> Sir Arther C. Clarke

Temporal annotation is difficult for both humans and machines. The task of determining how particular events are ordered or nested is part of this temporal annotation problem and has been the goal of this thesis. This is known as the temporal link labelling problem. The state of the art in this problem has advanced slowly in recent years, without reaching high enough performance levels to consider it solved. This thesis has investigated the problem of temporal link labelling.

A principled investigation began with a data-driven exploration of temporal links in a publicly-available corpus. This led to the identification of a set of difficult links, which many modern approaches cannot automatically label correctly. Formal and subjective analyses of this difficult link set were conducted. Results suggested multiple avenues of research (in the form of types of information seemingly used to label temporal links) and the two that were selected for investigation were signal-based links and links where there is a change of tense or aspect.

For the part of the signals, these were characterised as words or phrases associated with a pair of events or timexes that provide explicit information about their temporal relation. Experimentation with a machine learning approach showed that they were very helpful in link labelling, giving about a 50% error reduction. However, they are under-annotated in TimeBank, so attention turned to the task of automatically annotating signals. This was broken down into a two part task: discriminating signals (e.g. finding which phrases occur in text with a temporal sense and in a link labelling-supportive function) and association of signals, that is, determining which pair of events or timexes has its relation described by a given signal. Machine learning approaches and feature sets were identified for both these tasks. Finally, automatic signal annotation was

attempted on a corpus initially devoid of signals and the automatically-found signals used to help classifier-based temporal link labelling on that corpus, yielding an overall benefit compared to automatic labelling without any signal information.

To address the cases of tense shfits, Reichenbach's framework of tense was investigated. This included multiple interpretations from the framework to TimeML, including various mappings from tense and aspect pairs into its own tense structure. The framework proposes event and time orderings in simple and complex situations, based on a point-wise temporal logic. The framework also includes capacity for expression of abstract temporal points that is not present in TimeML. Initial validation suggested that the model could be of use for constraining the types of temporal relation between a given linked pair of event verbs. The model's output was added as a feature in a machine learning approach for temporal link labelling, and found to be of some utility in most cases. However, the problem of determining which events and times may be linked through this framework is open, and difficult to solve with existing tools. Critically, no existing resources are available in which this "temporal context" is annotated. A markup acting as an extension to TimeML is proposed for supporting this functionality, as well as supporting reasoning with and annotation for other aspects of Reichenbach's framework.

Overall, an investigation began with analysis of difficult temporal relations. Potential sources of information were identified that could be used to improve automatic system's performance when determining the types of these difficult relations. Of these, two were investigated – explicit temporal signals, and tense – and both exploited in such a way as to improve temporal relation typing. In the course of this exploitation a better understanding of discourse temporal relations and of both phenomena was reached, explained within this thesis.

7.1 Contributions

The work presented in this thesis furthered the understanding of some mechanisms used to convey temporal information in language. A full list of relevant publications and contributed resources is given in Appendix A.

7.1.1 Survey of Relations and Relation Typing Systems

Chapter 4 contained a data-driven analysis of temporal relation systems, in an attempt to first identify which relations are the hardest to automatically assign types to, and then to analysis this set of "difficult" links. TempEval-2 was an evaluation exercise where many systems attempted temporal relation labelling over a common data set. The exercise comprises the first analysis of the TempEval-2 participants' performance at relation-level, and the most in-depth analysis of any TempEval exercise.

As well as developing a definition of difficult links and defining a set of those links that are the hardest to automatically label within the TempEval-2 corpus, the chapter presents quantitative and qualitative analyses of the difficult link set. In this set, there were large groups of temporal links using explicit signals and others using tense shifts. These phenomena form the basis of the remainder of the thesis' investigation.

7.1.2 Temporal Signals

Chapter 5 investigated the role of explicit temporal signals in discourse, with regard to temporal relations. This chapter introduced a method for using signals to achieve a large relation typing performance boost on the temporal links that they co-ordinate. Seeing that signals can be useful, a characterisation of signals is presented, as well as a corpus survey of them. Finding under-annotation in TimeBank, temporal signal annotation guidelines are clarified and an augmented version of TimeBank including extra signals (and, as a result, some extra events, timexes and temporal links) is created. Given evidence for the utility of signals and high-quality ground truth data, the chapter turns to the automatic annotation of temporal signals. This annotation task is split into two sub-parts: signal discrimination (distinguishing temporal from non-temporal uses of signal words) and signal association (finding which timexes or events a given signal co-ordinates). Successful automatic methods for independent signal discrimination and signal association are introduced. These two sub-parts are then joined, in a joint annotation approach, and this approach for signal annotation evaluated, with satisfactory results. Finally, the question of the approach's ability to contribute to the overall temporal relation typing task is addressed. The joint approach is used to label signals and connect them to temporal relations. The results indicate an improvement in temporal relation labelling after this chapter's signal annotations are applied to a document.

7.1.3 Framework of Tense and Aspect

Building on the earlier analysis of difficult links, Chapter 6 introduces a theoretical framework for dealing with tense and aspect – that of Reichenbach (1947). This chapter first introduced tense and the framework, and suggested extensions to the framework to account for positional use of the speech point. Before applying the framework to the temporal relation typing task, it was rational to validate it. This was attempted using a minimal interpretation of the framework, with negative results. Failure analysis led to a new, advanced interpretation, including several novel concepts: an account of progressives; the notion of temporal context (groups within which certain tense rules can be applied); and the discovery that event-event relation typing based on tense suggests relations in semi-interval-link groupings. This advanced interpretation led to the first empirical validation of Reichenbach's framework of tense and aspect. Continuing, techniques for integrating the framework in supervised approaches for event-event and event-timex relation typing were introduced, giving slight benefits over the same approaches without information suggested by the framework. Problems were found with accurately automatically determining temporal context; a lack of context detection limits the applicability of the framework. The chapter closed with the description of a markup language for Reichenbach's framework, integrated with a current temporal annotation schema, in order to further research in this demonstrably promising area.

7.2 Future Work

The thesis suggests many directions of future work throughout. This section highlights some key points.

7.2.1 Sources of difficult links

The failure analysis of temporal relation typing given in Chapter 4 suggests a large nunmber of directions for further investigation. Only two of the problem areas discovered are explored in the rest of the thesis: signals and tense shifts. Many questions are raised about, for example, the impact that modality, iconicity, world knowledge and textual proximity have upon temporal relations. All these linguistic phenomena are worthy of further investigation, so that their rôle in temporal relation typing might be determined.

Recurrent is the theme of inference: the idea that the configuration of some temporal relations has a constraining impact on the possible configurations of other temporal relations. Temporal closure forms the basic part of this concept, but the role of temporal inference still remains largely unexplored. Approaches that attempt to use it often see only small improvements, though because global temporal constraint is difficult to perform, they have only included reduced-scope models of temporal inference. In an area full of noisy supervised learning output, it would be interesting to see a better integration of global temporal constraints. Prior work on temporal constraint networks (Dechter et al., 1991) has come close to this area. Techniques that can integrate the noisy, uncertain classifier output with global temporal constraints and discourse structure may yield new levels of temporal relation typing performance.

7.2.2 Temporal signals

While Chapter 5 introduced successful approaches for both annotating temporal signals and exploiting them for temporal relation typing, each of these approaches is a prototype and the first of its kind. There is certainly scope for improvement on each front.

Signal discrimination can be seen as a simplified word sense disambiguation (WSD) task: we are distinguishing temporal from non-temporal uses of expressions. While part-of-speech was shown not to be enough to determine whether or not a given signal was temporal, the approach taken still ignores the majority of the WSD literature (Navigli, 2009). For example, no context is taken into account when performing discrimination. Testing state-of-the-art WSD approaches on this binary classification task may lead to interesting results. Perhaps also the signal discrimination approach given in the chapter may contribute to some WSD tasks.

Signal association is a non-trivial task, and the approach given has some intrinsic limitations. For example, with the best-performing approach, only interval pairs within a certain number of sentences of each other are considered. This is shown by data from the corpus to already exclude some relations where the pair of intervals lie far apart. Other approaches to signal association, perhaps incorporating different discourse relations or some knowledge of pragmatics, may remove these boundaries and lead to increased performance.

Spatial and temporal signals are shown to have a lot in common. Spatial signals also seem to be critical in description of some spatial relations. It follows that the approach detailed in this thesis may be mapped without too much difficulty to the problem of automatically annotating spatial signals, and perhaps even to using them in automatic spatial relation typing (Kordjamshidi et al., 2011).

Finally, given the success of the signal annotation approach and the lack of signal annotation capability in current temporal annotation tools (e.g. Verhagen et al. (2005)), a next logical step is to package the techniques developed during the course of this thesis into a distributable tool for temporal signal annotation.

7.2.3 Reference time and temporal context

The work presented on Reichenbach's framework, and the new evaluation of its validity, progress many existing problems in computational linguistics concerning the management and interpretation of time in discourse. The chapter presents a big problem: that of determining temporal context. Clearly this is a direction for further work, marshalling current progress in discourse segmentation, syntactic analysis and the behaviour of temporal expressions. The results in this chapter suggest that automatically understanding temporal context permits accurate event-event and event-time relation typing.

However, temporal context is not the sole avenue for further research based on Reichenbach's framework. Multiple problems have called for a means of determining and reasoning with reference time. Aside from the temporal relation typing task, timex normalisation (interpreting an expression of a time) and story generation both require nuanced temporal reasoning, including awareness of the reference point.

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

The model is of use when generating language, for determining which tense to use. In fact, it is necessary to consider abstract temporal entities such as the reference point in order to know when to shift tense and how to properly describe events in other temporal frames of reference. A formal application of the model as it extends TimeML may prove useful to accurate language generation. Elson and McKeown (2010) describe how to relate events based on a "perspective" which is calculated from the reference and event times of an event pair. The authors construct a natural language generation system that requires accurate reference times in order to correctly write stories.

Portet et al. (2009) found reference point management critical to medical summary generation, in a situation where many small reports were generated with shifting speech and reference points, in order to helpfully unravel the meanings of tense shifts in minute-by-minute patient reports.

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

will be direct contributions to the completion of this task.

A computational model of the sequence of tenses may offer improvements in automatic machine translations. This is because accurately capturing temporal context permits more precise "analytical interlingual translation" (Horie et al., 2012).

There is also demand in journalism for changing a stock wire articles between present, past and anterior past, in order to suit a particular outlet's style guidelines. This mood switching can be accomplished using Reichenbach's framework.

Finally, the problem of datestamping documents automatically is not trivial. Reichenbach's framework provides the notion of speech time and means of bounding using permanence of the reference point between same-context events and attachment of events to fixed times via positional use of the reference point with a document's timexes. The model may therefore provide insights into this problem.

In summary, automatic determination of reference time for verbal expressions is an open problem, the solution of which is useful for a number of computational language processing tasks.

Appendix A

Resources and Publications

This appendix lists and briefly describes academic publications and language resources generated in the course of the PhD.

A.1 Publications

- Derczynski and Gaizauskas. 2010. "Analysing Temporally Annotated Corpora with CAVaT", Proceedings of the 7th Conference on International Language Resources and Evaluation.
- Derczynski. 2010. "Using Signals to Improve Automatic Classification of Temporal Relations", Proceedings of the ESSLLI Session, LNCS.
- Derczynski and Gaizauskas. 2010. "USFD2: Annotating Temporal Expresions and TLINKs for TempEval-2", Proceedings of the 5th Workshop on Semantic Evaluations, ACL.
- Derczynski and Gaizauskas. 2011. "An Annotation Scheme for Reichenbach's Verbal Tense Structure", Proceedings of the 6th Joint ACL - ISO Workshop on Interoperable Semantic Annotation.
- Derczynski and Gaizauskas. 2011. "RTMBank: A Corpus Annotated According to Reichenbach's Tense Model", Proceedings of Corpus Linguistics.
- Derczynski and Gaizauskas. 2011. "A Corpus-based Study of Temporal Signals", Proceedings of Corpus Linguistics.
- Burman, Jayapal, Kannan, Kavilikatta, Alhelbawy, Derczynski, Gaizauskas. 2011. "USFD at KBP 2011: Entity Linking, Slot Filling and Temporal Bounding", Proceedings of the Text Analysis Conference.
- Llorens, Derczynski, Saquete and Gaizauskas. 2012. "TIMEN: An Open Temporal Expression Normalization Resource", Proceedings of the 8th Conference on International Language Resources and Evaluation.

- Derczynski, Llorens and Saquete. 2012. "Massively Increasing TIMEX3 Resources: A Transduction Approach", Proceedings of the 8th Conference on International Language Resources and Evaluation.
- UzZaman, Llorens, Allen, Derczynski, Verhagen and Pustejovsky. 2012. "TempEval-3: Evaluating Events, Time Expressions, and Temporal Relations",- arXiv:1206.5333v1

A.2 Language Resources

Aside from experimental work, these language resources were generated in the course of this PhD.

A.2.1 CAVaT

CAVaT is an open source, modular checking utility for statistical analysis of features specific to temporally-annotated natural language corpora (Derczynski and Gaizauskas, 2010a). It provides reporting, highlights salient links between a variety of general and time-specific linguistic features, and also validates a temporal annotation to ensure that it is logically consistent and sufficiently annotated. Uniquely, CAVaT provides analysis specific to TimeML-annotated temporal information. CAVaT includes an API for loading TimeML documents into a portable database format, a command line query interface for interrogating and summarising TimeML data and a set of temporal consistency and evaluation checking tools.

A.2.2 RTMML

RTMML is a markup language for the tenses of verbs and temporal relations between verbs. There is a richness to tense in language that is not fully captured by existing temporal annotation schemata. Following Reichenbach we present an analysis of tense in terms of abstract time points, with the aim of supporting automated processing of tense and temporal relations in language. This allows for precise reasoning about tense in documents, and the deduction of temporal relations between the times and verbal events in a discourse. RTMML differs from TimeML (Pustejovsky et al., 2004) in that (1) it chiefly only annotates verbs that indicate events, (2) the information annotated about verbs is more nuanced, and (3) inter-verb links are defined using Reichenbach's three abstract points instead of event boundaries. RTMML has a syntax that can be adopted as an extension to TimeML. See also Section 6.6 and Derczynski and Gaizauskas (2011b).

A.2.3 TB-sig

TimeBank contains some signal annotations over text that describes the nature of a temporal link. However, these annotations are incomplete. TB-sig is a hand-curated version of TimeBank with improved signal annotations, adding extra timexes, events and temporal links where applicable. See also Section 5.5.

A.2.4 TempEval-2 analysis

TempEval-2 saw the comparison of diverse approaches to temporal link labelling over a fixed corpus and in a tightly controlled environment. This thesis includes an extensive formal analysis of the errors made by almost all the participants in the task, constituting the largest most recent survey of temporal link labelling efforts and difficulties. This is given in Section 4.2.

A.2.5 TIMEN

Automatically annotating temporal expressions is a research goal of increasing interest. Recognising them can be achieved with supervised machine learning, but interpreting them accurately (normalisation) is a complex task requiring human knowledge. TIMEN (Llorens et al., 2012a) is a community-driven tool for temporal expression normalisation. TIMEN is derived from current best approaches and is an independent tool, enabling easy integration in existing systems. It is argued that temporal expression normalisation can only be effectively performed with a large knowledge base and set of rules. Our solution is a framework and system with which to capture this knowledge for different languages.

A.2.6 T2T3 v.2

Saquete and Pustejovsky (2011) describe a technique for converting TIMEX2 to TIMEX3 annotations and present the T2T3 tool as an implementation of it. As some things annotated as TIMEX2s were no longer considered parts of temporal expressions in TimeML and instead assigned to other functions, T2T3 generates not only TIMEX3s but also extra TimeML elements. We upgraded the tool to better process complex TIMEX2s, using resources including temporal signals and mapping of Reichenbach's reference point (see Derczynski et al. (2012)).

A.2.7 TIMEX3 extended corpora

Applying T2T3 to earlier TIMEX2 corpora gave a 6x increase in the number of available TIMEX3s over the sum total of prior TIMEX3 resources. This extended dataset was useful for training state-of-the-art timex recognition tools and generated improved recognition accuracy. Timexes in the generated corpora were difficult to recognise using just existing TIMEX3 data, perhaps due to the limited variation of expression in previous newswire-only TIMEX3 data (Derczynski et al., 2012).

A.2.8 TempEval-3

Temporal annotation is a time-consuming task for humans, which has limited the size of annotated data in previous TempEval exercises. Current systems, however, are performing close to the inter-annotator reliability for entity recognition. This suggests that larger corpora could be built from automatically annotated data with minor human reviews. As part of TempEval-3, we explore whether there is value in adding a large automatically created silver standard to a hand-crafted gold standard. TempEval-3 differs from its ancestors in the following respects:

(i) size of the corpus: the dataset used comprises about 500K tokens of silver standard data and about 100K tokens of gold standard data for training, compared to the corpus of roughly 50K tokens corpus used in TempEval 1 and 2;

(ii) temporal relation task: the temporal relation classification tasks are to be performed from raw text, i.e. participants need to extract events and temporal expressions first, determine which ones to link and then obtain the relation types;

(iii) tasks not independent: participants must annotate temporal expressions and events in order to do the relation task;

(iv) temporal relation types: the full set of temporal interval relations in TimeML is used, rather than the reduced set used in earlier TempEvals;

(v) annotation: most of the corpus was automatically annotated by the state-of-the-art systems from TempEval-2, a portion of the corpus, including the test dataset, that is human reviewed;

(vi) evaluation: we will report a temporal awareness score for evaluating temporal relations, to help to rank systems with a single score.

This will be the largest temporal link labeling exercise to date, in terms of datasets available; see also UzZaman et al. (2012).

Appendix B

Annotated Corpora and Annotation Tools

B.1 Introduction

TimeML is a standard for annotating time in natural language. It includes annotations for the lexicalised entities TIMEX3, EVENT and SIGNAL, and for the abstract entities TLINK, SLINK, ALINK and MAKEINSTANCE. The syntax is XML-like, with inline annotation. For the temporal link labelling task, one is interested in TIMEX3, EVENT, SIGNAL and TLINK. The MAKEINSTANCE tag gives events extra information and instantiates them for use in TLINKs, and so also contains useful information. TimeML has recently become a non-free ISO standard ISO-TimeML, which incorporates a few changes to event description and permits stand-off annotation. As almost all prior work and all existing resources use TimeML or an extension thereof, this thesis considers only TimeML in general.

B.2 Corpora

B.2.1 TimeBank

TimeBank is a human annotated TimeML corpus of 183 newswire texts. TimeBank v1.2 contains 6 418 TLINKs, 1 414 TIMEX3s and 7 935 EVENTs, and is 3004kB in size. This is tiny compared to some other types of corpus, but is large enough to be useful and has been battered enough by the community through a few versions to be considered robust. TimeBank's creation (Pustejovsky et al., 2003) involved a large human annotator effort and a few different versions (Boguraev et al., 2007); it is currently the largest temporally annotated corpus.

TimeBank 1.2 contains 183 documents, comprising about 64 000 tokens. Over these tokens are:

- 7935 EVENTs
- 6418 TLINKs

TimeML tag	Exact match IAA
TIMEX3	0.83
EVENT	0.78
TLINK	0.55

Table B.1: Inter-annotator agreement in TimeBank v1.2; data from (Boguraev et al., 2007)

Table B.2: Distribution of TIMEX3 type

TIMEX3 type	Frequency	Proportion
DATE	1164	82.3%
DURATION	175	12.4%
TIME	63	4.46%
SET	12	0.849%
Total	1414	

- 7940 INSTANCEs
- 688 SIGNALs
- 1414 TIMEX3s
- 2932 SLINKs
- 265 ALINKs

Table B.3: Distribution of TIMEX3 mod

TIMEX3 mod	Frequency	Proportion
START	28	30.4%
APPROX	16	17.4%
END	16	17.4%
EQUAL_OR_LESS	8	8.7%
MID	7	7.61%
EQUAL_OR_MORE	6	6.52%
LESS_THAN	4	4.35%
MORE_THAN	3	3.26%
ON_OR_AFTER	3	3.26%
BEFORE	1	1.09%
None	0	0.0%
Total	92	

Table B.4: Distribution of EVENT class

EVENT class	**Frequency**	**Proportion**
OCCURRENCE	4215	53.1%
STATE	1117	14.1%
REPORTING	1028	13.0%
I_ACTION	681	8.58%
I_STATE	584	7.36%
ASPECTUAL	262	3.3%
PERCEPTION	48	0.605%
Total	7935	

Table B.5: Distribution of EVENT pos

EVENT pos	**Frequency**	**Proportion**
VERB	5122	64.5%
NOUN	2225	28.0%
OTHER	299	3.77%
ADJECTIVE	266	3.35%
PREPOSITION	28	0.353%
Total	7940	

The remainder of this subsection presents summary information over the events, timexes, signals and and temporal relations in TimeBank 1.2.

Table B.6: Distribution of EVENT modality

EVENT modality	**Frequency**	**Proportion**
would	127	39.7%
could	49	15.3%
may	31	9.69%
can	26	8.13%
none	21	6.56%
might	16	5.0%
must	14	4.38%
should	13	4.06%
have to	5	1.56%
'd	2	0.625%
possible	2	0.625%
should have to	2	0.625%
close	1	0.313%
delete	1	0.313%
depending on	1	0.313%
have_to	1	0.313%
having to	1	0.313%
likelihood	1	0.313%
potential	1	0.313%
to	1	0.313%
unlikely	1	0.313%
until	1	0.313%
would have to	1	0.313%
would_be	1	0.313%
None	0	0.0%
Total	320	

Table B.7: Distribution of EVENT polarity

EVENT polarity	**Frequency**	**Proportion**
POS	7651	96.4%
NEG	289	3.64%
Total	7940	

Table B.8: Distribution of TLINK reltype

TLINK reltype	**Frequency**	**Proportion**
BEFORE	1408	21.9%
IS_INCLUDED	1357	21.1%
AFTER	897	14.0%
IDENTITY	743	11.6%
SIMULTANEOUS	671	10.5%
INCLUDES	582	9.07%
DURING	302	4.71%
ENDED_BY	177	2.76%
ENDS	76	1.18%
BEGUN_BY	70	1.09%
BEGINS	61	0.95%
IAFTER	39	0.608%
IBEFORE	34	0.53%
DURING_INV	1	0.0156%
Total	6418	

A r1 B	B r2 C														
	BE	**AF**	**IN**	**II**	**DU**	**SI**	**IA**	**IB**	**ID**	**BG**	**EN**	**BB**	**EB**	**DI**
BEFORE	BE	X	BE	X	BE	BE	X	BE	BE	BE	X	BE	BE	BE
AFTER	X	AF	AF	X	AF	AF	AF	X	AF	X	AF	AF	AF	AF
INCLUDES	X	X	IN	X	IN	IN	X	X	IN	X	X	IN	IN	IN
IS_INCLUDED	BE	BE	X	II	II	II	AF	BE	II	II	II	X	X	II
DURING	BE	AF	IN	II	SI	SI	IA	IB	SI	BG	EN	BB	EB	SI
SIMULTANEOUS	BE	AF	IN	II	SI	SI	IA	IB	SI	BG	EN	BB	EB	SI
IAFTER	X	AF	AF	X	IA	IA	AF	X	IA	X	IA	AF	IA	IA
IBEFORE	BE	X	BE	X	IB	IB	X	BE	IB	IB	X	IB	BE	IB
IDENTITY	BE	AF	IN	II	SI	SI	IA	IB	ID	BG	EN	BB	EB	SI
BEGINS	BE	AF	X	II	BG	BG	IA	BE	BG	BG	II	X	X	BG
ENDS	BE	AF	X	II	EN	EN	AF	IB	EN	II	EN	X	X	EN
BEGUN_BY	BE	AF	IN	X	BB	BB	IA	X	BB	X	X	BB	IN	BB
ENDED_BY	BE	X	IN	X	EB	EB	X	IB	EB	X	X	X	EB	EB
DURING_INV	BE	AF	IN	II	SI	SI	IA	IB	SI	BG	EN	BB	EB	SI

Table B.9: Transitivity table for the TimeML relation set; an X indicates that no clear inference can be made. Abbreviations: BE = Before, AF = After, IN = Includes, II = Is_included, DU = During, SI = Simultaneous, IA = Iafter, IB = Ibefore, ID = Identity, BG = Begins, EN = Ends, BB = Begun_by, EB = Ended_by, DI = During_inv.

B.2.2 AQUAINT

The second-largest English TimeML corpus is the AQUAINT TimeML corpus. The AQUAINT TimeML corpus consists of around 80 TimeML-annotated newswire documents. These are grouped by the story that they cover, with each group related to the same story, reporting progress on events through time.

Due to repeated text and heavy event co-reference, the AQUAINT corpus requires some care to use correctly. One must firstly maintain document level testing and training set separation, to ensure that evaluation examples are not those found verbatim in training data. Further, due to the corpus' repeated attention to the same story over multiple documents, some event summaries and orderings are repeated using the same text across documents. For this reason, it is best to split datasets by story, so that the background summaries repeated in articles on the same story do not contaminate test and training data. Finally, separately from text re-use, there is re-description of events using later knowledge. Because the news stories contain information on the same topic describing the same events, it is important not to include later articles in the training set for a classifier evaluated on articles published prior. That is to say, evaluation should not be performed using articles that the training data provides hindsight over. This is a common constraint with time-series data (Bergmeir and Benítez, 2012) and applies to this TimeML corpus because of its repeated coverage of the same super-events.

B.2.3 Other TimeML corpora

There have been other TimeML corpora released, in a range of languages, including French (Bittar et al., 2011), Italian (Caselli et al., 2011) and Romanian (Forăscu et al., 2007).

B.2.4 Other Non-TimeML corpora

The TempEval corpora (Verhagen et al., 2009, 2010) feature event, timex and tlink annotations over non-parallel news text in multiple different languages. The set of TLINKs is slightly different from those available in TimeML, being simpler and including a `VAGUE` relation. TempEval-2 included English, Spanish, French, Italian, Chinese and Korean.

The ACE (Automatic Content Extraction) exercises were based on purpose-built corpora that included a large number of TIMEX2 annotations, comprising almost 26 000 TIMEX2s. For comparison, TimeBank has only 1 414 TIMEX3 annotations.

The WikiWars corpora (Mazur and Dale, 2010; Strötgen and Gertz, 2011) are derived from WikiPedia articles about wars. These articles tend to contain temporal expressions of a variety of granularities and forms and a generally quite long pieces of connected prose. WikiWars and WikiWars-DE are both annotated according to TIMEX2 and are resources of significant size.

B.3 Temporal annotation tools

Temporal annotation is a complex task for humans; to this end, we have annotation guidelines to simplify things. Typing XML is also a rather painful experience for us, let alone a specific variant

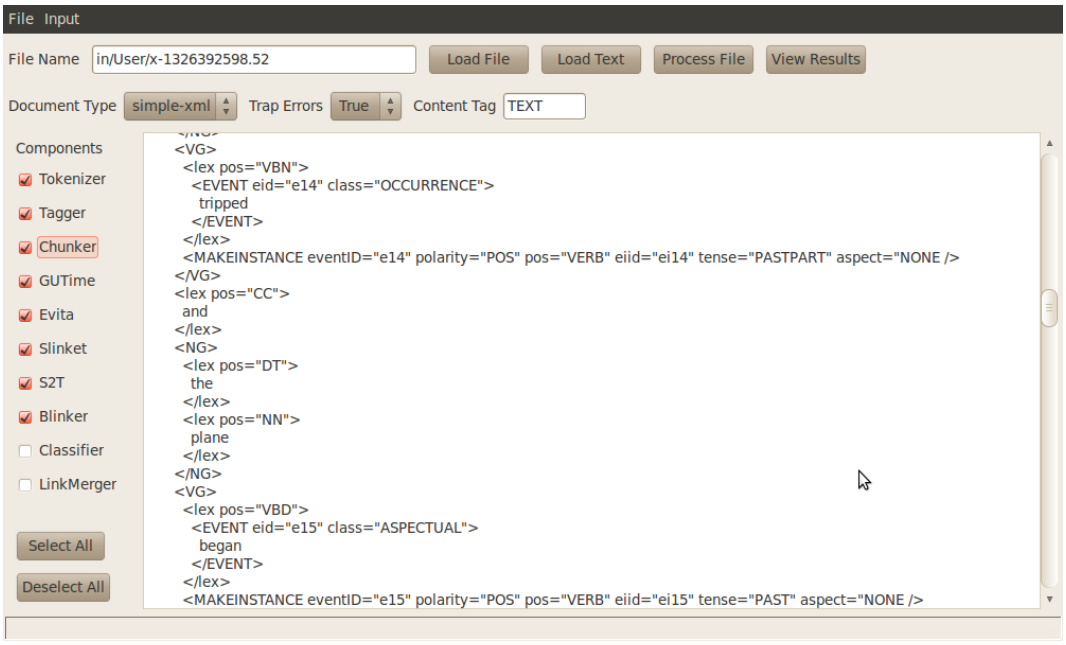

Figure B.1: Automatically annotating text with TTK.

of it that captures abstract information, such as TimeML; and to this end, we have temporal annotation tools that can simplify the task.

In this section, we will first describe TARSQI, a state-of-the-art toolkit containing many components for temporal annotation of text. We will then discuss the problem of visually presenting temporal information.

B.3.1 TARSQI/TTK

A set of tools for automatic TimeML annotation are bundled together in the form of the TARSQI toolkit, TTK (Verhagen and Pustejovsky, 2008), which is described as "a modular system for automatic temporal and event annotation of natural language texts" (Figure B.1). TTK adopts a multi-stage work-flow, beginning with the entry of raw unannotated text, followed by automated annotation and then user correction of machine-produced results. The toolkit ties together a large number of components, including EVITA (Saurí et al., 2005), Slinket (Saurí et al., 2006; Saurı et al., 2006), SputLink (Verhagen, 2005) and TBOX (Verhagen, 2007), using a plugin-based Python framework. It is easy for users to see which plugins have been involved in annotation decisions, making TTK useful for analyzing individual components.

As well as identifying and annotating events and times, TTK also includes sophisticated logic for labelling TLINKs. As far as rule-based relation identification goes, S2T (Verhagen and Pustejovsky, 2008; Verhagen, 2004) is capable of generating TLINKs from SLINKs and Blinker – based on GutenLink (Verhagen et al., 2005) – contains a large set of relation postulations given configurations of EVENTs and TIMEXs and focuses on TLINKs.

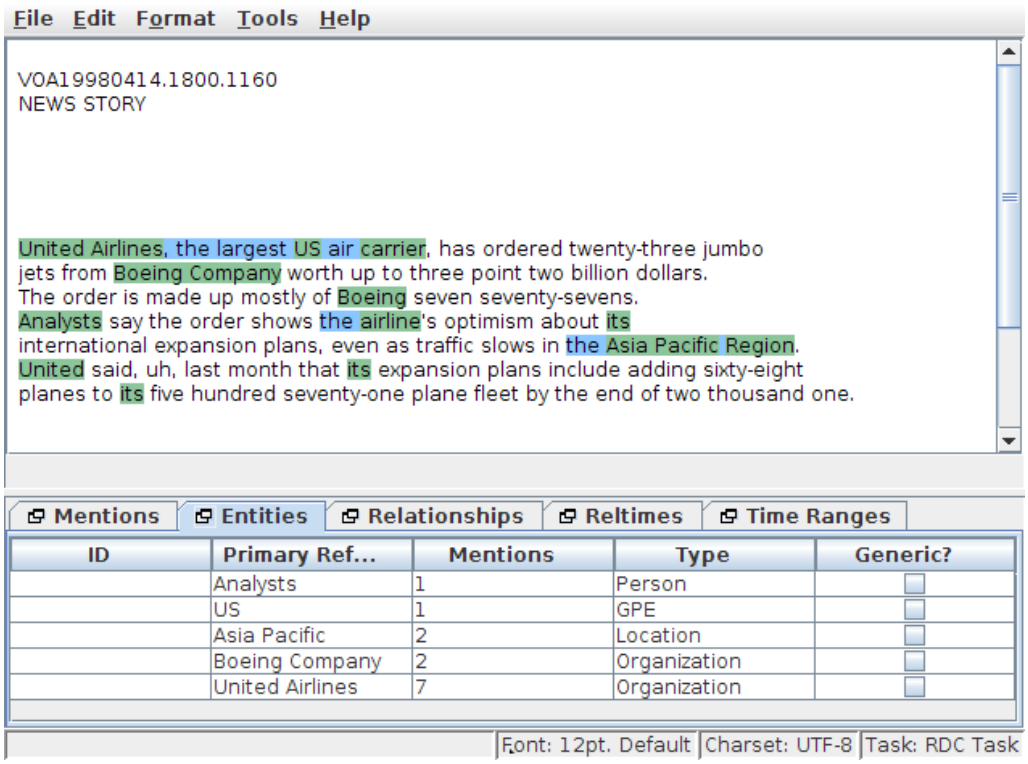

Figure B.2: Manually annotating text with Callisto.

Instead of prior versions of the toolkit which permitted co-operation of link annotation components via a voting mechanism (Verhagen et al., 2005). TTK has a separate Link Merger component. The merger uses confidence scores from individual components as well as a pre-set bias (for example, to give low priority to the large number of classifier-generated links) to order candidate links. These are then sequentially tested against a temporal graph of the discourse, with consistency checking between each addition; inconsistent links are not added. This makes it impossible to revoke possibly incorrect information once it has been added, but generates a consistent annotation where high confidence is at least partially rewarded.

B.3.2 Callisto / Tango

TANGO is an assistive annotation tool that helps users build correct annotations from suggestions made by the included automatic temporal annotation systems, as well as a visual representation component. It is integrated within Callisto (Figure B.2), a general-purpose manual linguistic annotation tool. Callisto's TANGO component for TimeML annotation is particularly strong for ease of temporal link annotation.

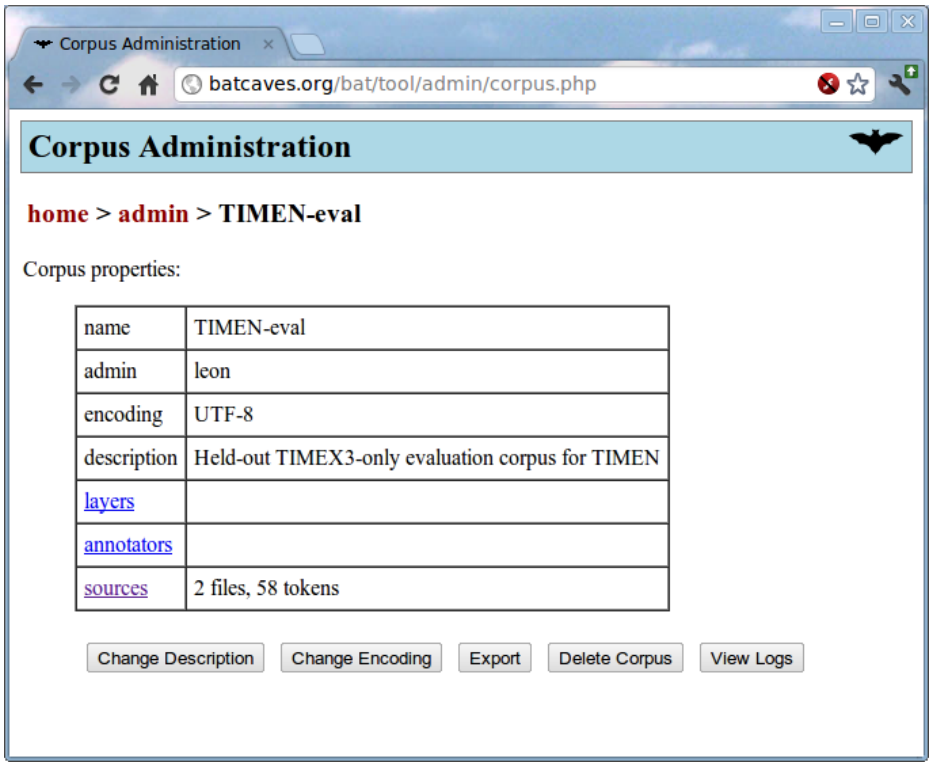

Figure B.3: Overseeing a BAT annotation project.

B.3.3 BAT

The Brandeis Annotation Tool, or BAT (Verhagen, 2010), enables collaborative semantic annotation and breaks down annotation into subtasks. It is a web-based tool, with administrator overview (see Figure B.3). Multiple asynchronous and concurrent annotations can be made, making BAT a flexible tool for co-ordinating gold standard TimeML annotations. It has been used to create the TempEval-2 and Ita-TimeBank datasets.

B.3.4 Other tools

Other purpose-built tools exist, such as Dante (Mazur and Dale, 2009) which concentrates on temporal expression tagging and normalisation across many genres of text but is not are publicly available. Existing general purpose language toolkits may also be adapted to cater for TimeML processing, such as NLTK (Loper and Bird, 2002), GATE (Cunningham et al., 2002, 2013) and Xara (Burnard and Todd, 2003).

Appendix C

RTMML Reference

This appendix details extensions made to TimeML, so that it may capture extra information helpful for temporal reasoning, based upon Reichenbach's framework of tense and aspect Reichenbach (1947).

C.1 Examples

C.1.1 Fiction

From *David Copperfield* by Charles Dickens:

(C.1) *When he had put up his things for the night he took out his flute, and blew at it, until I almost thought he would gradually blow his whole being into the large hole at the top, and ooze away at the keys.*

We give RTMML for the first five verbal events from Example C.1 RTMML in Figure C.1. The fifth, `v5`, exists in a context that is instantiated by `v4`; its reference time is defined as such.

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

Figure C.1: RTMML for a passage from David Copperfield.

We can use one `link` element to show that `v2`, `v3` and `v4` all use the same reference time as `v1`. The temporal relation between event times of `v1` and `v2` can be inferred from their shared reference time and their tenses; that is, given that `v1` is anterior past and `v2` simple past, we know $E_{v1} < R_{v1}$ and $E_{v2} = R_{v2}$. As our `<rtmlink>` states $R_{v1} = R_{v2}$, then $E_{v1} < E_{v2}$. Finally, `v5` and `v6` happen in the same context, described with a second SAME_TIMEFRAME link.

C.1.2 Editorial news

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

(C.2) *Saddam appeared to accept a border demarcation treaty he had rejected in peace talks following the August 1988 cease-fire of the eight-year war with Iran.*

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

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

(C.3)
```
Saddam <EVENT eid="e74" class="I_STATE">
appeared</EVENT> to accept a border demarcation treaty he had <EVENT eid="e77"
class="OCCURRENCE">rejected</EVENT>

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

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

C.1.3 Linking events to calendar references

RTMML makes it possible to precisely describe the nature of links between verbal events and times, via positional use of the reference point. We will link an event to a temporal expression,

and suggest a calendrical reference for that expression, allowing the events to be placed on a calendar. Consider the below text, from wsj_0533.tml in TimeBank.

(C.4) *At the close of business Thursday, 5,745,188 shares of Connaught and C$44.3 million face amount of debentures, convertible into 1,826,596 common shares, had been tendered to its offer.*

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

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

C.2 Annotation notes

As can be seen in Table 6.2, there is not a one-to-one mapping from English tenses to the nine specified by Reichenbach. In some annotation cases, it is possible to see how to resolve such ambiguities. Even if view and tense are not clearly determinable, it is possible to define relations between S, E and R; for example, for arrangements corresponding to the simple future, $S < E$. In cases where ambiguities cannot be resolved, one may annotate a disjunction of relation types; in this example, we might say "$S < R$ or $S = R$" with `sr="<="`.

Contexts seem to have a shared speech time, and the $S - R$ relationship seems to be the same throughout a context. Sentences which contravene this (e.g. *"By the time I ran, John will have arrived"*) are rather awkward. Contexts are typically broken by timexes (e.g. positional use of the reference point), shifting of frame of reference by use of *"then"*, use of temporal signals or any boundary of reported speech (e.g. starting and ending quotes).

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

Appendix D

CAVaT Reference

This section contains a reference guide for the CAVaT package (Derczynski and Gaizauskas, 2010a). Up to date information can always be found at http://cavat.googlecode.com.

D.1 Installation and configuration

The first time CAVaT is run, it will attempt to create a directory `$HOME$/.cavat/`, where it will store its SQLite files.

D.2 Getting started

Enter the following to load a TimeML corpus into the "test" database - it's important to include the trailing slash `/` in the path:

```
cavat> corpus import /home/user/corpus/data/ to test
```

Depending on your disk and CPU speeds, this might take about 10-20 seconds per megabyte of TimeML. If it seems to take longer, you can get more information aabout what CAVaT is doing during import by enabling debug mode before import:

```
cavat> debug on
```

Leave debug mode with a simple:

```
cavat> debug off
```

Once the corpus has loaded, you can use `corpus info` to see metadata about the import, or `corpus list` to see an available list of corpora. To switch between corpora, and to select a newly loaded one, enter `corpus use <name>`.

D.3 Queries

The show command is used for generating reports on the currently loaded corpus. Reports focus on one tag type, and give information about their attributes. One can view all values for a tag

with `list` reports, or the distribution of values with `distribution` reports, or simply see how many instances of that tag list a value for a field with `state` reports.

Reports can be provided in multiple formats; there is:

- `screen` - for screen or fixed-width font output
- `csv` - comma separated values
- `tex` - LaTeX table format

The general format for report generation is:

```
show <report type> of <tag> <field> [as <format>]
```

To try a simple query, enter:

```
cavat> show distribution of tlink reltype
```

You should see a table listing the values listed for relType in the current corpus' TLINK tags, as well as their frequencies. To see how many TLINKs use a signal, and use the result in a LaTeX document, you can try:

```
cavat> show state of tlink signalid as tex
```

Bibliography

Ahn, D., S. Adafre, and M. Rijke (2005), "Towards task-based temporal extraction and recognition." In *Dagstuhl Seminar Proceedings*, volume 5151. 15, 128

Allen, J. F. (1984), "Towards a general theory of action and time." *Artificial intelligence*, 23, 123–154. 17

Allen, J. F. and P. J. Hayes (1989), "Moments and points in an interval-based temporal logic." *Computational Intelligence*, 5, 225–238. 25

Allen, J. (1983), "Maintaining knowledge about temporal intervals." *Communications of the ACM*, 26, 832–843. 17, 22, 23, 36

Ando, R. (2004), "Exploiting unannotated corpora for tagging and chunking." In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, Association for Computational Linguistics Morristown, NJ, USA. 13, 19

Androutsopoulos, I. (1999), "Temporal meaning representations in a natural language front-end." *Arxiv preprint cs/9906020*. 18

ARDA (2006), "Aquaint timeml corpus." URL http://www.timeml.org/site/timebank/timebank.html. 40

Bergmeir, C. and J. Benítez (2012), "On the use of cross-validation for time series predictor evaluation." *Information Sciences*, 191, 192–2132. 171

Bestgen, Y. and W. Vonk (1999), "Temporal adverbials as segmentation markers in discourse comprehension." *Journal of Memory and Language*, 42, 74–87. 3, 14, 75

Bethard, S. and J. Martin (2007), "Cu-tmp: temporal relation classification using syntactic and semantic features." In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, 129–132, Association for Computational Linguistics, Stroudsburg, PA, USA, URL http://dl.acm.org/citation.cfm?id=1621474.1621499. 47

Bethard, S., J. Martin, and S. Klingenstein (2007a), "Finding temporal structure in text: Machine learning of syntactic temporal relations." *International Journal of Semantic Computing*, 1, 441. 42, 43, 44

Bethard, S., J. Martin, and S. Klingenstein (2007b), "Timelines from Text: Identification of Syntactic Temporal Relations." In *Proceedings of the International Conference on Semantic Computing*, 11–18. 10, 35, 42, 47, 81

Bies, A., M. Ferguson, K. Katz, R. MacIntyre, V. Tredinnick, G. Kim, M. Marcinkiewicz, and B. Schasberger (1995), "Bracketing guidelines for Treebank II style Penn Treebank project." *University of Pennsylvania*. 97

Bittar, A., P. Amsili, P. Denis, and L. Danlos (2011), "French TimeBank: an ISO-TimeML annotated reference corpus." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, 130–134, Association for Computational Linguistics. 171

Blaheta, D. and E. Charniak (2000), "Assigning function tags to parsed text." In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, 240, Morgan Kaufmann Publishers Inc. 97

Boguraev, B. and R. Ando (2005), "TimeML-compliant text analysis for temporal reasoning." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*. 12, 13, 16, 19, 39, 40, 43, 47, 151

Boguraev, B., J. Pustejovsky, R. Ando, and M. Verhagen (2007), "TimeBank Evolution as a Community Resource for TimeML Parsing." *Language Resources and Evaluation*, 41, 91–115. 35, 165, 166

Bramsen, P., P. Deshpande, Y. Lee, and R. Barzilay (2006), "Finding temporal order in discharge summaries." In *AMIA Annual Symposium Proceedings*, volume 2006, 81, American Medical Informatics Association. 3, 47

Brée, D., A. Feddag, and I. Pratt (1993), "Towards a formalization of the semantics of some temporal prepositions." *Time & Society*, 2, 219. 75

Brée, D. and R. Smit (1986), "Temporal relations." *Journal of Semantics*, 5, 345. 75

Bruce, B. (1972), "A model for temporal references and its application in a question answering program." *Artificial intelligence*, 3, 1–25. 23

Buckland, M. and F. Gey (1994), "The relationship between recall and precision." *Journal of the American society for information science*, 45, 12–19. 13

Burman, A., A. Jayapal, S. Kannan, M. Kavilikatta, A. Alhelbawy, L. Derczynski, and R. Gaizauskas (2011), "Usfd at kbp 2011: Entity linking, slot filling and temporal bounding." In *Proceedings of the Text Analytics Conference*. 45

Burnard, L. and T. Todd (2003), "Xara: an XML aware tool for corpus searching." In *Proceedings of Corpus Linguistics 2003*, 142–4. 174

By, T. (2002), *Tears in the Rain*. Ph.D. thesis, University of Sheffield. 131

Carlson, A., J. Betteridge, R. Wang, E. Hruschka Jr, and T. Mitchell (2010), "Coupled semi-supervised learning for information extraction." In *Proceedings of the third ACM international conference on Web search and data mining*, 101–110. 1

Caselli, T., V. Lenzi, R. Sprugnoli, E. Pianta, and I. Prodanof (2011), "Annotating events, temporal expressions and relations in Italian: the It-TimeML experience for the Ita-TimeBank." *ACL HLT 2011*, 143. 171

Chambers, N. and D. Jurafsky (2008a), "Jointly combining implicit constraints improves temporal ordering." In *Proceedings of the EMNLP*, 698–706, ACL. 41, 44, 45, 47

Chambers, N. and D. Jurafsky (2008b), "Unsupervised learning of narrative event chains." *Proceedings of ACL-08, Hawaii, USA*. 34

Chambers, N., S. Wang, and D. Jurafsky (2007), "Classifying temporal relations between events." In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, 173–176, Association for Computational Linguistics. 45, 47

Charniak, E. (2000), "A maximum-entropy-inspired parser." In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, 132–139, Morgan Kaufmann Publishers Inc. 96

Cheng, Y., M. Asahara, and Y. Matsumoto (2007), "NAIST.Japan: temporal relation identification using dependency parsed tree." In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, 245–248, Association for Computational Linguistics. 43, 47

Chinchor, N. and P. Robinson (1997), "MUC-7 named entity task definition." In *Proceedings of the 7th Message Understanding Conference*. 16

Chklovski, T. and P. Pantel (2004a), "Path Analysis for Refining Verb Relations." In *Proceedings of KDD Workshop on Link Analysis and Group Detection (LinkKDD-04)*. 34

Chklovski, T. and P. Pantel (2004b), "Verbocean: Mining the web for fine-grained semantic verb relations." In *Proceedings of EMNLP*, volume 4, 33–40. 34

Cohen, D. and S. Schwer (2012), "Proximal deixis with calendar terms: Cross-linguistic patterns of temporal reference." *ms. submitted in lingua*. 14

Costa, F. and A. Branco (2012), "Aspectual type and temporal relation classification." In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 266–275. 143

Cunningham, H., D. Maynard, K. Bontcheva, and V. Tablan (2002), "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications." In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, URL http://www.gate.ac.uk/. 174

Cunningham, H., V. Tablan, A. Roberts, and K. Bontcheva (2013), "Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics." *PLoS computational biology*, 9, e1002854. 174

Dang, H., J. Lin, and D. Kelly (2008), "Overview of the trec 2006 question answering track." In *Proceedings of the Text Retrieval and Evaluation Conference*. 45

De Marneffe, M., B. MacCartney, and C. Manning (2006), "Generating typed dependency parses from phrase structure parses." In *Proceedings of the International Conference on Language Resources and Evaluation*. 109, 147

Dechter, R., I. Meiri, and J. Pearl (1991), "Temporal constraint networks." *Artificial intelligence*, 49, 61–95. 158

Denis, P. and P. Muller (2010), "Comparison of different algebras for inducing the temporal structure of texts." In *Proceedings of the 23rd International Conference on Computational Linguistics*, 250–258, Association for Computational Linguistics. 23

Denis, P. and P. Muller (2011), "Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition." In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. 27, 32

Derczynski, L. and R. Gaizauskas (2010a), "Analysing Temporally Annotated Corpora with CAVaT." In *Proceedings of the Language Resources and Evaluation Conference*, 398–404. 27, 76, 96, 98, 162, 179

Derczynski, L. and R. Gaizauskas (2010b), "USFD2: Annotating temporal expresions and TLINKs for TempEval-2." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 337–340, Association for Computational Linguistics. 44, 47, 55

Derczynski, L. and R. Gaizauskas (2010c), "Using signals to improve automatic classification of temporal relations." In *Proceedings of the ESSLLI StuS*. 7, 43, 44, 47, 78

Derczynski, L. and R. Gaizauskas (2011a), "A Corpus-based Study of Temporal Conjunctions." In *Proceedings of Corpus Linguistics*. 7, 75, 82

Derczynski, L. and R. Gaizauskas (2011b), "An Annotation Scheme for Reichenbach's Verbal Tense Structure." In *Workshop on Interoperable Semantic Annotation*, 10–17. 7, 150, 162

Derczynski, L., H. Llorens, and E. Saquete (2012), "Massively increasing TIMEX3 resources: a transduction approach." In *Proceedings of the Language Resources and Evaluation Conference*. 15, 163

Derczynski, L. and R. Gaizauskas (2013a), "Empirical Validation of Reichenbach's Tense Framework." In *Proceedings of the 10th Conference on Computational Semantics*, 71–82, ACL. 7, 138, 144

Derczynski, L. and R. Gaizauskas (2013b), "Temporal Signals Help Label Temporal Relations." In *Proceedings of the annual meeting of the Association for Computational Linguistics*, ACL. 78

Derczynski, L., H. Llorens, and N. UzZaman (2013), "TimeML-strict: clarifying temporal annotation." *arXiv preprint arXiv:1304.7289*. 17

Diessel, H. (2008), "Iconicity of sequence: A corpus-based analysis of the positioning of temporal adverbial clauses in English." *Cognitive linguistics*, 19, 465–490. 67

Dorr, B. and T. Gaasterland (2006), "Summarization-inspired temporal-relation extraction: tense-pair templates and treebank-3 analysis." Technical Report CS-TR-4844, University of Maryland, College Park, MD, USA. 85

Dowty, D. (1986), "The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics?" *Linguistics and philosophy*, 9, 37–61. 130

Eddington, A. (1928), *The nature of the physical world*. Macmillen. 124

Elson, D. and K. McKeown (2010), "Tense and Aspect Assignment in Narrative Discourse." In *Proceedings of the Sixth International Conference in Natural Language Generation*. 159

Ferro, L., L. Gerber, I. Mani, B. Sundheim, and G. Wilson (2005), "Tides 2005 standard for the annotation of temporal expressions." Technical Report 03-1046, The MITRE Corporation. 15, 17, 151

Filatova, E. and V. Hatzivassiloglou (2004), "Event-based extractive summarization." In *Proceedings of the ACL Workshop on Summarization*, 104–111. 3

Fillmore, C. (1971), *Lectures on deixis*. CSLI Publications Stanford, California. 125

Forăscu, C., R. Ion, and D. Tufiş (2007), "Semi-automatic Annotation of the Romanian TimeBank 1.2." In *Proceedings of the RANLP 2007 Workshop on Computer-aided language processing-CALP*, volume 30, 1–7. 171

Freksa, C. (1992), "Temporal reasoning based on semi-intervals." *Artificial intelligence*, 54, 199–227. 26, 46

Freund, Y. and R. Schapire (1996), "Experiments with a new boosting algorithm." In *Machine Learning: Proceedings of the Thirteenth International Conference*, 148–156. 101

Freund, Y. and R. Schapire (1997), "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of Computer and System Sciences*, 55, 119–139. 101

Gabbard, R., M. Marcus, and S. Kulick (2006), "Fully parsing the Penn Treebank." In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 184–191, Association for Computational Linguistics. 97

Gaizauskas, R., H. Harkema, M. Hepple, and A. Setzer (2006), "Task-oriented extraction of temporal information: The case of clinical narratives." In *Temporal Representation and Reasoning, 2006. TIME 2006. Thirteenth International Symposium on*, 188–195, IEEE. 47

Gaizauskas, R. and Y. Wilks (1998), "Information Extraction: Beyond Document Retrieval." *Journal of Documentation*, 54, 70–105. 1

Galton, A. (2008), "Temporal logic." In *The Stanford Encyclopedia of Philosophy* (E. Zalta, ed.), The Metaphysics Research Lab, Stanford University. 25

Giorgi, A. and F. Pianesi (1997), *Tense and aspect: From semantics to morphosyntax*. Oxford University Press, USA. 132

Goranko, V., A. Montanari, and G. Sciavicco (2004), "A road map of interval temporal logics and duration calculi." *Journal of Applied Nonclassical Logics*, 14, 9–54. 23, 25, 27

Grover, C., R. Tobin, B. Alex, and K. Byrne (2010), "Edinburgh-LTG: TempEval-2 system description." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 333–336, Association for Computational Linguistics. 20

Grünwald, P. (1996), "A minimum description length approach to grammar inference." *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, 203–216. 71

Gusev, A., N. Chambers, K. D.R., P. Khaitan, S. Bethard, and D. Jurafsky (2011), "Using query pattens to learn the duration of events." In *Proceedings of the 9th International Conference on Computational Semantics*, 145–154. 9

Ha, E., A. Baikadi, C. Licata, and J. Lester (2010), "Ncsu: Modeling temporal relations with markov logic and lexical ontology." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, 341–344, Association for Computational Linguistics. 43, 47, 55

Hagège, C. and X. Tannier (2007), "Xrce-t: Xip temporal module for tempeval campaign." In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, 492–495. 42, 47

Han, B. (2009), "Reasoning about a Temporal Scenario in Natural Language." In *Proceedings of the International Joint Conferences on Artificial Intelligence*. 18

Han, B., D. Gates, and L. Levin (2006), "From language to time: A temporal expression anchorer." In *Proceedings of the 13th International Symposium on Temporal Representation and Reasoning (TIME 2006)*. 15, 18, 159

Harris, R. and W. Brewer (1973), "Deixis in memory for verb tense." *Journal of Verbal Learning and Verbal Behavior*, 12, 590–597. 67, 68, 122

Hepple, M., A. Setzer, and R. Gaizauskas (2007), "USFD: preliminary exploration of features and classifiers for the TempEval-2007 tasks." In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, 438–441, Association for Computational Linguistics. 40, 47, 122

Hinrichs, E. (1986), "Temporal anaphora in discourses of english." *Linguistics and Philosophy*, 9, 63–82. 132

Hitzeman, J. (1997), "Semantic partition and the ambiguity of sentences containing temporal adverbials." *Natural Language Semantics*, 5, 87–100. 14, 75

Hitzeman, J. (2005), "Text type and the position of a temporal adverbial within the sentence." In *Proceedings of the 2005 international conference on Annotating, extracting and reasoning about time and events*, 29–40, Springer-Verlag. 75

Ho-Dac, L. and M. Péry-Woodley (2008), "Temporal adverbials and discourse segmentation revisited." In *Multidisciplinary Approaches to Discourse*. 75

Hobbs, J. R. and F. Pan (2004), "An ontology of time for the semantic web." *ACM Transactions on Asian Language Information Processing (TALIP)*, 3, 66–85. 14, 23

Horie, A., K. Tanaka-Ishii, and M. Ishizuka (2012), "Verb temporality analysis using Reichenbach's tense system: Towards interlingual MT." In *Proceedings of the International Conference on Computational Linguistics*, 471–482, ACL. 3, 160

Hornstein, N. (1990), *As time goes by: Tense and universal grammar*. MIT Press. 129, 131, 134

Hovy, E., M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel (2006), "Ontonotes: the 90% solution." In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, 57–60, Association for Computational Linguistics. 46

Howald, B. and E. Katz (2011), "On the explicit and implicit spatiotemporal architecture of narratives of personal experience." *Spatial Information Theory*, 434–454. 4, 45

Hristova, D. (2006), "The neoreichenbachian model of tense syntax and the rusian active participles." *Harvard Ukrainian Studies*, 28, 155–164. 132

Huggett, N. (2010), "Zeno's paradoxes." In *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), winter 2010 edition. 124

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

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

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

Jean-Louis, L., R. Besançon, and O. Ferret (2010), "Using Temporal Cues for Segmenting Texts into Events." *Advances in Natural Language Processing*, 150–161. 3

Ji, H., R. Grishman, H. Dang, X. Li, K. Griffit, and J. Ellis (2011), "Overview of the TAC2011 Knowledge Base Population Track." In *Proceedings of the Text Analytics Conference*. 45

Johansson, R., A. Berglund, M. Danielsson, and P. Nugues (2005), "Automatic text-to-scene conversion in the traffic accident domain." In *International Joint Conference on Artificial Intelligence*, volume 19, 1073. 4

Jung, H., J. Allen, N. Blaylock, W. de Beaumont, L. Galescu, and M. Swift (2011), "Building timelines from narrative clinical records: Initial results based-on deep natural language understanding." *ACL HLT 2011*, 146. 4

Kelley Jr, J. and M. Walker (1959), "Critical-path planning and scheduling." *AFIPS Joint Computer Conferences*, 160–173. 37

Kiss, T. and J. Strunk (2006), "Unsupervised multilingual sentence boundary detection." *Computational Linguistics*, 32, 485–525. 96, 108, 145

Klein, D. and C. Manning (2003), "Fast exact inference with a factored model for natural language parsing." *Advances in neural information processing systems*, 15, 3–10. 96, 97, 108

Klein, W. (1994), *Time in language*. Germanic linguistics, Routledge, London [u.a.]. 125

Kolya, A., A. Ekbal, and S. Bandyopadhyay (2011), "Event-time relation identification using machine learning and rules." In *Text, Speech and Dialogue*, 117–124, Springer. 41, 47

Kolya, A., A. Ekbal, and S. Bandyopadhyay (2010a), "Event-event relation identification: A crf based approach." In *Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on*, 1–8, IEEE. 42, 47

Kolya, A., A. Ekbal, and S. Bandyopadhyay (2010b), "Ju_cse_temp: A first step towards evaluating events, time expressions and temporal relations." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 345–350. 42, 47

Kordjamshidi, P., M. Van Otterlo, and M. Moens (2011), "Spatial role labeling: Towards extraction of spatial relations from natural language." *ACM Transactions on Speech and Language Processing (TSLP)*, 8, 4. 158

Kowalski, R. and M. Sergot (1989), "A logic-based calculus of events." In *Foundations of knowledge base management*, 23–55, Springer. 26, 138

Lapata, M. and A. Lascarides (2006), "Learning sentence-internal temporal relations." *Journal of Artificial Intelligence Research*, 27, 85–117. 47, 96, 109, 122

Lee, C. and G. Katz (2009), "Error analysis of the tempeval temporal relation identification task." *SEW-2009 Semantic Evaluations: Recent Achievements and Future Directions*, 138–145. 48, 55

Lee, K. and L. Romary (2010), "Towards Interoperability of ISO Standards for Language Resource Management." In *International Conference on Global Interoperability for Language Resources*. 151

Levenshtein, V. (1965), "Binary codes capable of correcting spurious insertions and deletions of ones." *Problems of Information Transmission*, 1, 8–17. 37

Lin, D. and P. Pantel (2002), "Discovery of inference rules for question-answering." *Natural Language Engineering*, 7, 343–360. 34

Lintean, M. and V. Rus (2007), "Naive bayes and decision trees for function tagging." In *FLAIRS Conference*, 604–609. 97

Llorens, H. (2011), *A Semantic Approach to Temporal Information Processing*. Ph.D. thesis, University of Alicante. 14

Llorens, H., L. Derczynski, E. Saquete, and R. Gaizauskas (2012a), "TIMEN: An Open Temporal Expression Normalization Resource." In *Proceedings of the Language Resources and Evaluation Conference*. 20, 163

Llorens, H., E. Saquete, and B. Navarro (2010), "TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2." In *Proceedings of SemEval-2010*, 284–291, ACL. 20, 43, 47, 55

Llorens, H., E. Saquete, and B. Navarro (2011), "Syntax-motivated context windows of morpho-lexical features for recognizing time and event expressions in natural language." In *Natural Language Processing and Information Systems*, 295–299, Springer. 20

Llorens, H., E. Saquete, and B. Navarro-Colorado (2012b), "Automatic system for identifying and categorizing temporal relations in natural language." *International Journal of Intelligent Systems*, 680–708. 43

Loper, E. and S. Bird (2002), "NLTK: The natural language toolkit." In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, 63–70, Association for Computational Linguistics, URL http://www.nltk.org/. 174

Lyons, J. (1977), *Semantics, volume 2*. Cambridge University Press. 124

Mani, I., J. Hitzeman, J. Richer, D. Harris, R. Quimby, and B. Wellner (2008), "SpatialML: Annotation scheme, corpora, and tools." In *Proceedings of LREC*, volume 8. 87

Mani, I., J. Pustejovsky, and R. Gaizauskas (2005), *The Language of Time: A Reader*. Oxford University Press. 126

Mani, I. and B. Schiffman (2005), "Temporally anchoring and ordering events in news." *Time and Event Recognition in Natural Language*. 9

Mani, I., B. Schiffman, and J. Zhang (2003), "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*, 55–57, Association for Computational Linguistics. 41

Mani, I., M. Verhagen, B. Wellner, C. Lee, and J. Pustejovsky (2006), "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*, 760, Association for Computational Linguistics. 29, 40, 43, 78, 79

Mani, I., B. Wellner, M. Verhagen, and J. Pustejovsky (2007), "Three approaches to learning TLINKS in TimeML." Technical Report CS-07-268, Brandeis University, Waltham, MA, USA. 10, 32, 34, 36, 40, 41, 42, 47, 70

Mani, I. and G. Wilson (2000), "Robust temporal processing of news." In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 69–76, Association for Computational Linguistics. 15, 20, 159

Marcus, M., G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger (1994), "The Penn Treebank: annotating predicate argument structure." In *Proceedings of the workshop on Human Language Technology*, 114–119, Association for Computational Linguistics. 97

Marcus, M., M. Marcinkiewicz, and B. Santorini (1993), "Building a large annotated corpus of English: The Penn Treebank." *Computational linguistics*, 19, 330. 98

Marsic, G. (2011), *Temporal Processing of News: Annotation of Temporal Expressions, Verbal Events and Temporal Relations*. Ph.D. thesis, University of Wolverhampton. 43, 47

Matlock, T., M. Ramscar, and L. Boroditsky (2005), "On the experiential link between spatial and temporal language." *Cognitive Science*, 29, 655–664. 124

Mazur, P. and R. Dale (2008), "What's the date? High accuracy interpretation of weekday." In *22nd International Conference on Computational Linguistics (Coling 2008), Manchester, UK*, 553–560. 19, 177

Mazur, P. and R. Dale (2009), "The DANTE Temporal Expression Tagger." In *Human Language Technology. Challenges of the Information Society*, 257, Springer-Verlag. 174

Mazur, P. and R. Dale (2010), "WikiWars: A new corpus for research on temporal expressions." In *Proceedings of the EMNLP*, 913–922. 159, 171

Mazur, P. and R. Dale (2011), "LTIMEX: Representing the Local Semantics of Temporal Expressions." In *Proceedings of the 1st International Workshop on Advances in Semantic Information Retrieval (ASIR)*, 201–208. 19

McDermott, D. (1982), "A temporal logic for reasoning about plans and actions." *Cognitive Science*, 6, 101–155. 25

McTaggart, J. (1908), "The unreality of time." *Mind*, 17, 457. 2, 124

Michaelis, L. (2006), "Time and tense." *The Handbook of English Linguistics*, 220–243. 124

Min, C., M. Srikanth, and A. Fowler (2007), "LCC-TE: A hybrid approach to temporal relation identification in news text." In *Proceedings of SemEval-2007*, 219–222, ACL. 42, 47

Minsky, M. (1991), "Logical versus analogical or symbolic versus connectionist or neat versus scruffy." *AI magazine*, 12, 34. 41

Mirroshandel, S. and G. Ghassem-Sani (2010), "Temporal relations learning with a bootstrapped cross-document classifier." In *Proceeding of the 19th European Conference on Artificial Intelligence*, 829–834. 45, 47

Mirroshandel, S. and G. Ghassem-Sani (2011), "Temporal relation extraction using expectation maximization." In *Proceedings of RANLP*. 45

Mirroshandel, S., G. Ghassem-Sani, and M. Khayyamian (2010), "Using syntactic-based kernels for classifying temporal relations." *Journal of Computer Science and Technology*, 26, 68–80. 41, 45, 47

Mirroshandel, S., G. Ghassem-Sani, and A. Nasr (2011), "Active learning strategies for support vector machines, application to temporal relation classification." In *Proceedings of 5th International Joint Conference on Natural Language Processing*, 56–64. 45

Moens, M. and M. Steedman (1988), "Temporal ontology and temporal reference." *Computational linguistics*, 14, 15–28. 9, 23

Musillo, G. and P. Merlo (2005), "Assigning function labels to unparsed text." In *Proceedings of RANLP'05*. 97

Navigli, R. (2009), "Word Sense Disambiguation: a survey." *ACM Computing Surveys*, 41, 1–69. 96, 158

Pan, F., R. Mulkar, and J. Hobbs (2006), "Learning event durations from event descriptions." In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 393–400, Association for Computational Linguistics. 9

Paslawska, A. and A. van Stechow (2003), "Perfect readings in russian." *Perfect Explorations*, 2, 307. 131

Portet, F., E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, and C. Sykes (2009), "Automatic generation of textual summaries from neonatal intensive care data." *Artificial Intelligence*, 173, 789–816. 159

Pratchett, T. (1986), *The Light Fantastic*. Colin Smythe. 123

Prior, A. (1967), *Past, Present and Future*. Clarendon. 133

Prior, A. (1968), "Tense logic and the logic of earlier and later." *Papers on Time and Tense*, 116–134. 23

Puşcaşu, G. (2007a), "Discovering temporal relations with tictac." In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*. 47

Puşcaşu, G. (2007b), "WVALI: Temporal relation identification by syntactico-semantic analysis." In *Proceedings of the 4th International Workshop on SemEval*, volume 2007, 484–487. 43, 47

Pustejovsky, J. (1991), "The syntax of event structure." *Cognition*, 41, 47. 2, 10

Pustejovsky, J. (2009). Personal correspondence. 29

Pustejovsky, J., P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, et al. (2003), "The Timebank Corpus." In *Corpus Linguistics*, 647–656. 40, 165, 176

Pustejovsky, J., B. Ingria, R. Sauri, J. Castano, J. Littman, and R. Gaizauskas (2004), "The Specification Language TimeML." In *The Language of Time: A Reader*, 545–557, Oxford University Press. 10, 11, 16, 17, 162

Pustejovsky, J., R. Knippen, J. Littman, and R. Saurí (2005), "Temporal and Event Information in Natural Language Text." *Language Resources and Evaluation*, 39, 123–164. 17

Pustejovsky, J., K. Lee, H. Bunt, and L. Romary (2010), "ISO-TimeML: An International Standard for Semantic Annotation." In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. 17

Quinlan, J. (1993), *C4. 5: programs for machine learning*. Morgan Kaufmann. 100

Quirk, R., S. Greenbaum, G. Leech, J. Svartvik, and D. Crystal (1985), *A comprehensive grammar of the English language*, volume 1. Longman. 85

Reichenbach, H. (1947), "The tenses of verbs." In *Elements of Symbolic Logic*, Dover Publications. 41, 93, 122, 149, 157, 175

Rissanen, J. (1978), "Modeling by shortest data description." *Automatica*, 14, 465–471. 71

Ritter, A., O. Etzioni, S. Clark, et al. (2012), "Open domain event extraction from twitter." In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1104–1112, ACM. 10

Saquete, E. (2010), "Terseo+t2t3 transducer: a systems for recognizing and normalizing timex3." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 317–320, Association for Computational Linguistics. 20

Saquete, E. and J. Pustejovsky (2011), "Automatic transformation from TIDES to TimeML annotation." *Language Resources and Evaluation*. 163

Saurí, R., R. Knippen, M. Verhagen, and J. Pustejovsky (2005), "Evita: a robust event recognizer for qa systems." In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 707, Association for Computational Linguistics. 13, 172

Saurı, R., M. Verhagen, and J. Pustejovsky (2006), "Annotating and recognizing event modality in text." In *The 19th International FLAIRS Conference, FLAIRS 2006*. 172

Saurí, R., M. Verhagen, and J. Pustejovsky (2006), "Slinket: A partial modal parser for events." In *Language Resources and Evaluation Conference, LREC*. 172

Savova, G., S. Bethard, W. Styler, J. Martin, M. Palmer, J. Masanz, and W. Ward (2009), "Towards temporal relation discovery from the clinical narrative." In *AMIA Annual Symposium Proceedings*, volume 2009, 568, American Medical Informatics Association. 4

Schlüter, N. (2001), "Temporal specification of the present perfect: a corpus-based study." *Language and Computers*, 36, 307–315. 75

Setzer, A. (2001), *Temporal information in newswire articles: an annotation scheme and corpus study*. Ph.D. thesis. 22

Setzer, A. and R. Gaizauskas (2000), "Annotating events and temporal information in newswire texts." In *Proceedings of the Second International Conference On Language Resources And Evaluation (LREC-2000), Athens, Greece*, volume 31. 32, 76

Setzer, A. and R. Gaizauskas (2001), "A pilot study on annotating temporal relations in text." In *Proceedings of the workshop on Temporal and spatial information processing-Volume 13*, 1–8, Association for Computational Linguistics. 22

Setzer, A., R. Gaizauskas, and M. Hepple (2005), "The role of inference in the temporal annotation and analysis of text." *Language Resources and Evaluation*, 39, 243–265. 23, 29, 35, 36

Shannon, C. (1949), "Communication theory of secrecy systems." *Bell system technical journal*, 28, 656–715. 71

Song, F. and R. Cohen (1988), "The interpretation of temporal relations in narrative." In *Proceedings of the 7th National Conference of AAAI*. 126

Steedman, M. (1982), "Reference to past time." *Speech, Place and Action*, 125–157. 10

Stevenson, M. and Y. Wilks (2005), "Word sense disambiguation." *The Oxford Handbook of Comp. Linguistics*, 249–265. 96

Stocker, K. (2012), "The time machine in our mind." *Cognitive Science*. 124

Strötgen, J. and M. Gertz (2010), "HeidelTime: High quality rule-based extraction and normalization of temporal expressions." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 321–324, Association for Computational Linguistics. 20, 159

Strötgen, J. and M. Gertz (2011), "WikiWarsDE: A German Corpus of Narratives Annotated with Temporal Expressions." In *Proceedings of the Conference of the German Society for Computational Linguistics and Language Technology (GSCL 2011)*, 129–134, Hamburg, Germany. 171

Tanaka, K. (1990), "On Reichenbach's Approach to Tense." *Tsukuba English Studies*, 9, 61–75. 133

Tannier, X. and P. Müller (2011), "Evaluating temporal graphs built from texts via transitive reduction." *Journal of Artificial Intelligence Research*, 40, 375–413. 35, 37

Tatu, M. and M. Srikanth (2008), "Experiments with reasoning for temporal relations between events." In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, 857–864, Association for Computational Linguistics. 44, 47

Tsang, E. (1987), "The consistent labeling problem in temporal reasoning." In *Proceedings of the AAAI Conference*, 251–255, AAAI Press. 25

UzZaman, N. and J. Allen (2010), "TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 276–283, Association for Computational Linguistics. 20, 43, 47, 55

UzZaman, N. and J. Allen (2011), "Temporal evaluation." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 351–356, Association for Computational Linguistics. 37

UzZaman, N., H. Llorens, J. Allen, L. Derczynski, M. Verhagen, and J. Pustejovsky (2012), "Tempeval-3: Evaluating events, time expressions, and temporal relations." *arXiv*, abs/1206.5333. 37, 38, 164

van Rijsbergen, C. (1979), *Information retrieval, Second edition*. Butterworths. 13

Vasilakopoulos, A. and W. Black (2005), "Temporally ordering event instances in natural language texts." In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*. 41, 47

Vendler, Z. (1957), "Verbs and times." *The philosophical review*, 66, 143–160. 11, 150

Verhagen, M. (2004), *Times Between The Lines*. Ph.D. thesis, Brandeis University. 27, 110, 172

Verhagen, M. (2005), "Temporal closure in an annotation environment." *Language Resources and Evaluation*, 39, 211–241. 33, 35, 172

Verhagen, M. (2007), "Drawing TimeML Relations with TBox." *Lecture Notes in Computer Science*, 4795, 7. 172

Verhagen, M. (2010), "The Brandeis Annotation Tool." In *Language Resources and Evaluation Conference, LREC*, volume 2010, 3638–3643. 174

Verhagen, M., R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky (2007), "Semeval-2007 task 15: Tempeval temporal relation identification." In *SemEval-2007: 4th International Workshop on Semantic Evaluations*. 35, 36, 42

Verhagen, M., R. Gaizauskas, F. Schilder, M. Hepple, J. Moszkowicz, and J. Pustejovsky (2009), "The TempEval Challenge: identifying temporal relations in text." *Language Resources and Evaluation*, 43, 161–179. 38, 44, 171

Verhagen, M., I. Mani, R. Sauri, R. Knippen, S. Jang, J. Littman, A. Rumshisky, J. Phillips, and J. Pustejovsky (2005), "Automating temporal annotation with TARSQI." In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, 84, Association for Computational Linguistics. 159, 172, 173

Verhagen, M. and J. Pustejovsky (2008), "Temporal Processing with the TARSQI Toolkit." In *CoLing 2008: Posters and Demonstrations*, 189–192. 172

Verhagen, M., R. Saurí, T. Caselli, and J. Pustejovsky (2010), "SemEval-2010 task 13: TempEval-2." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 57–62, Association for Computational Linguistics. 5, 12, 38, 44, 53, 91, 171

Vilain, M. and H. Kautz (1986), "Constraint propagation algorithms for temporal reasoning." In *Proceedings of the Fifth National Conference on Artificial Intelligence*, 377–382. 25, 29, 36

Vlach, F. (1993), "Temporal adverbials, tenses and the perfect." *Linguistics and Philosophy*, 16, 231–283. 75

Wang, W., J. Su, and C. Tan (2010), "Kernel based discourse relation recognition with temporal ordering information." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 710–719, Association for Computational Linguistics. 3

Yoshikawa, K., S. Riedel, M. Asahara, and Y. Matsumoto (2009), "Jointly identifying temporal relations with Markov logic." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 405–413, Association for Computational Linguistics. 42, 44, 45, 47

Zhang, T., F. Damerau, and D. Johnson (2002), "Text chunking based on a generalization of winnow." *The Journal of Machine Learning Research*, 2, 615–637. 39

Zipf, G. (1935), *The psycho-biology of language*. Houghton-Mifflin. 83

Index