Temporal Dependency Structure Modeling

A Dissertation

Presented to

The Faculty of the Graduate School of Arts and Sciences
Brandeis University

Department of Computer Science

Nianwen Xue, Advisor

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

by

Yuchen Zhang

February 2020

The signed version of this form is on file in the Graduate School of Arts and Sciences.

This dissertation, directed and approved by Yuchen Zhang's Committee, has been accepted and approved by the Faculty of Brandeis University in partial fulfillment of the requirements for the degree of:

DOCTOR OF PHILOSOPHY

Eric Chasalow, Dean
Graduate School of Arts and Sciences

Dissertation Committee:

Nianwen Xue, Computer Science
James Pustejovsky, Computer Science
Marc Verhagen, Computer Science
Dan Roth, Computer and Information Science, University of Pennsylvania

Acknowledgements

I would like to thank my advisor Nianwen (Bert) Xue for his support and encouragement during my PhD years. He taught me not only scientific knowledge and technical skills, but also life experiences and philosophies. Always patient with me, he witnessed and helped my growth as a researcher as well as an adult human being. I'm very grateful for having him as my advisor and for having been able to work with him for the past years.

I would like to thank my dissertation committee members Nianwen (Bert) Xue, James Pustejovsky, Marc Verhagen, and Dan Roth for their time and help, insightful feedback and inspiring advice on my work. I want to give a special thanks to Marc for helping me with details on writing in this dissertation.

I would like to thank my supportive lab mates: Yaqin Yang, Te Rutherford, Chuan Wang, Jiarui Yao, and Jayeol Chun. I appreciate their moral and intellectual supports through the years.

I would like to thank my parents for raising me and for the education they provided me before graduate school. This PhD work wouldn't be possible without these early investments. I would like to thank my husband Jeff for being a great friend and companion through the years. No matter I'm going through smooth moments or struggles, he was always there for me. Always loving and supportive, he helped me find the courage, strength, and wisdom to keep going, and also reminded me to look around at life. He made me regret no past, fear no future, and so grateful for the present. Finally, I would like to thank my parents in law, for loving me just the way I am.

ABSTRACT

Temporal Dependency Structure Modeling

A dissertation presented to the Faculty of the
Graduate School of Arts and Sciences of Brandeis University
Waltham, Massachusetts

by Yuchen Zhang

An important task in understanding the meaning of natural language text is to represent and understand the temporal information in the text. Time expressions, events that happened at some time points, and temporal relations between these time expressions and events are the three basic temporal information commonly present in texts. A well designed machine-readable temporal representation is crucial for representing and understanding these information efficiently. Most fundamental research on temporal information modeling has been representing temporal relations in a pair-wise manner – the temporal relation between pairs of time expressions and/or events are explicitly and separately modeled. This stream of representations faces a few challenges. First, human annotation for this representation is laborious and on some level arbitrary. Second, computation on this representation is expensive and inefficient on scalability. Third, due to the nature of temporal transitivity,annotations (human or computational) harbor potential conflicts on temporal relations.

In this dissertation, we introduce a new temporal representation to address these challenges – the Temporal Dependency Tree (TDT) structure. A Temporal Dependency Tree represents temporal information in a text as a single dependency tree. Time expressions and events are represented as nodes on the tree, while temporal relations are represented as edges. A TDT explicitly models n temporal relations for a text with n time expressions and events, reducing human annotation labor, computation complexity, and temporal transitivity conflicts. As a proof-of-concept, we performed annotation experiments on the TDT representation to show stable and high inter-annotator

agreements. To support further linguistic study on TDT and automatic system training, we built an expert-annotated TDT corpus (on two domains: news and narratives). One step closer to automatic temporal information modeling and understanding, we built a competitive Temporal Dependency Parser that parses time expressions and events in a text into a Temporal Dependency Tree structure. Finally, to collect larger amount of TDT data more efficiently, and further support the training of better temporal dependency parsers, we experimented with crowdsourcing approaches and built a TDT corpus with high agreements through crowdsourcing.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivations

Natural Language Understanding (NLU) aims at understanding the meaning of natural language text. One important component of the meaning of a text is the temporal information in it. Recognizing time expressions, anchoring events on a timeline, and understanding the temporal relations between events and temporal expressions are some fundamental parts of understanding the meaning of a text. Moreover, automatic detection of such temporal information is important to many downstream applications in Natural Language Processing (NLP) and Artificial Intelligence (AI) that include but are not limited to story timeline construction, question answering, text summarization, information extraction, and others.

To enable computers to automatically extract and work with such temporal information, a machine readable temporal representation is necessary. The dominant approach in prior work on temporal information modeling adopts a pair-wise approach: for pairs of time expressions and/or events

in a text, the system models whether there exists a temporal relation between them and what the relation is. This temporal relation is selected from a pre-defined finite set of temporal relation categories. For example, for the following three events A, B, and C, a pair-wise approach models all three pairs of temporal relations: B *before* A, A *before* C, and B *before* C.

(1): A. John went into the florist shop.

B. He had promised Mary some flowers.

C. He picked out three red roses, two white ones and one pale pink

Representative work in this vein includes TimeML [Pustejovsky et al., 2003a], a rich temporal relation markup language that is based on and extends Allen's Interval Algebra [Allen, 1984]. TimeML has been further enriched and extended for annotation in other domains [O'Gorman et al., 2016, Styler IV et al., 2014a, Mostafazadeh et al., 2016]. Corpora annotated with these schemes [Pustejovsky et al., 2003b, O'Gorman et al., 2016] are shown to have stable Inter-Annotator Agreements, validating the temporal relations proposed in the TimeML.

Accordingly, automatic systems working with this pair-wise representation usually solves a classification problem: given an individual pair of time expressions and/or events, the system predicts whether they are temporally related and which specific relation holds between them. Representative pair-wise temporal relation classification systems include the participating systems in a series of TempEval shared tasks [Verhagen et al., 2007a, Verhagen et al., 2010a, UzZaman et al., 2012, Bethard et al., 2015a, Bethard et al., 2016a, Bethard et al., 2017] and others [Bethard et al., 2007, Chambers et al., 2007].

As shown in the previous example, one inherent issue with the pair-wise representation is redundancy. From B *before* A and A *before* C, we can naturally infer that B is *before* C without explicitly modeling this temporal relation. Unfortunately, redundancy leaves room for conflicts.

As each pair of event is independently classified, it opens the door for annotators or automatic systems to produce conflicting temporal relations within a text. One such scenario is if the human annotator or the system annotates the relations between the three pairs as B *before* A, A *before* C, and B ***after*** C. Another way to view pair-wise models is as a graph-based representation with potential cycles. And temporal relations on a cycle in this graph can conflict with each other (e.g. A *before* B, B *before* C, and A ***after*** C).

One possible way to alleviate such conflicts in automatic predictions is by enforcing global constraints to ensure temporal transitivity [Verhagen, 2004, Chambers and Jurafsky, 2008a, Ning et al., 2018a, Allen, 1984]. However, applying global constraints requires a fully connected graph. Namely, given a text with n time expressions and events, $\binom{n}{2}$ temporal relations need to be explicitly annotated. This approach will quickly get impractical with longer texts for human annotators, making it hard to produce such training data for machine learning based systems.

To solve the issues of redundancy and conflicts in a pair-wise temporal representation, and to reduce the computational complexity of global constraints, we propose a new temporal representation in which temporal information in text is modeled as a dependency structure. More specifically, this structure is a single-rooted dependency tree for the entire text. Time expressions and events are represented as nodes in the tree, and temporal relations are represented as edges between them. We call this representation a Temporal Dependency Tree (TDT) Structure. Figure 1.1 gives a small example text and its Temporal Dependency Tree structure, where time expressions are represented as orange nodes, events as green nodes, some pre-defined meta blue nodes (see more at §3), and temporal relations on edges.

Prior work on structured models for temporal information in text include the Temporal Discourse Model (TDM) for narrative structures [Mani and Pustejovsky, 2004], the Narrative Container

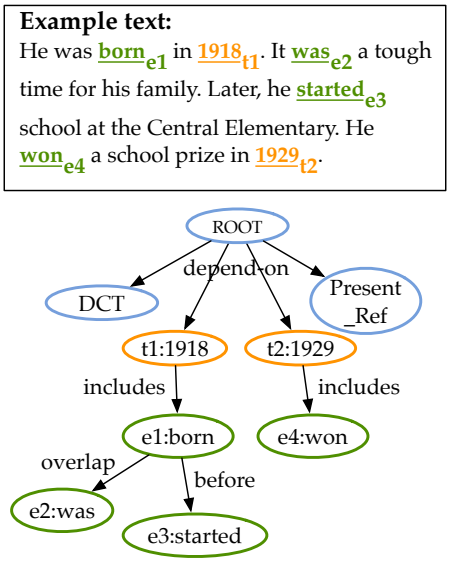


Figure 1.1: Example text and temporal dependency tree. Meta nodes are shown in blue, time expressions in orange, and events in green. TDT also includes meta nodes "Past_Ref," "Future_Ref," and "Atemporal" which are not shown here.

model for temporal information [Pustejovsky and Stubbs, 2011], the Temporal Dependency Structure for narrative events [Bethard et al., 2012], and the Multi-axis Annotation Scheme for Event Temporal Relations [Ning et al., 2018b]. These temporal models are described in details in 2.2.3.1, and comparisons between our proposed TDT structure and these models are discussed in details in 2.2.4.

Before going into details of the design of a TDT structure (which will be fully described in Chapter 3), we would like to first briefly summarize its potential benefits as follows. For TDT annotation, since an annotator does not have to annotate all pairs of events and time expressions in a text, annotating long texts becomes practical. Given a text with n time expressions and events, only n temporal relations need to be explicitly modeled in a TDT. Moreover, TDT guarantees an acyclic tree structure, which leaves no room for cyclic temporal conflicts. From the point of view of linguistic annotation, this alleviates potential inconsistencies when annotators pick a subset of $\binom{n}{2}$

relations to annotate using each individual's judgement. From a computational perspective, a TDT eliminates potential conflicts in predicted temporal relations (more on this in Chapter 3). Although a TDT is not modeling all possible pairs of temporal relations, additional temporal relations can still be inferred along the paths of the temporal dependency tree or through the ordering of time expressions. And just as pair representations and graph representations, a dependency tree is also a very well studied structure in NLP. It's amenable to a wide range of parsing algorithms, and is easy to use in downstream applications.

The Temporal Dependency Structure has its roots from the computational linguistic research on on Temporal Anaphora [Reichenbach, 1947, Partee, 1973, Partes, 1984, Hinrichs, 1986, Webber, 1988, Bohnemeyer, 2009]. In research on temporal anaphora, a temporal relation is modeled as an anaphoric relation where an event or time expression is the antecedent of another event or time expression (the anaphor). The antecedent is called the **Reference Time** of the anaphor. And the temporal location of the anaphor can only be interpreted with respect to its antecedent (i.e. its reference time). When applying this theory practically to our data, we define the antecedent of an anaphor time expression or event as the reference time with respect to which the temporal location of the anaphor **can be most precisely determined**. With this definition, there will only be one antecedent for each anaphor (i.e. one reference time for each time expression or event). Then, by representing a (reference time, anaphor) pair as a (parent node, child node) pair on the tree, we will arrive at a TDT structure that naturally fulfills the formal requirements of a valid dependency tree.

Literature on Temporal Anaphora is reviewed in Section 2.1, and detailed specifications on Temporal Dependency Tree Structure are introduced in Chapter 3. As a proof-of-concept and also in order to facilitate research on automatic TDT parsers, we developed annotation guidelines for TDT and annotated a TDT corpus in Chinese. To compare the different temporal structures between different domains, this corpus covers articles from two domains: news reports and narrative fairy

tales. Having proved through our annotation experiments that TDT is an intuitive structure that can be annotated with high inter-annotator agreements, we further developed automatic parsers for TDT. We hope these parsers will benefit downstream applications by providing structural temporal information.

The annotation process of our TDT corpus consists of several rounds of annotator training before the actual annotation effort was carried out, and an expert adjudication pass at the end. Annotator training and expert annotation provide high quality data at the cost of time and expense. Therefore, we are interested in developing an approach to crowdsourcing TDT annotations. Crowdsourcing is usually used to collect data on relatively straightforward tasks such as speech transcription, "copying online info into a form", or "identifying a smiley face". Compared to most crowdsourcing tasks, a temporal dependency tree structure is a very complex concept, and to collect TDT annotations through crowdsourcing is a very challenging task. Therefore, we studied the feasibility of using crowdsourcing to collect TDT annotations, built an English TDT corpus through crowdsourcing with high Inter-Annotator Agreements, and experimented with parsers on this English TDT data.

In summary, this thesis introduces a structured representation for temporal information in text – the Temporal Dependency Tree Structure, presents data collection effort for this structure, introduces the first statistical parsers for this structure, and provides analysis and discussions on collected annotations and trained parsers.

1.2 Contributions

The main contributions of this thesis are summarized in this section.

Structured Interpretation of Temporal Relations

This thesis describes our research on interpreting temporal relations in text in a structured manner. In this research, we developed a structured representation for temporal relations in text – the Temporal Dependency Tree Structure .

First of all, we designed refined classifications for time expressions and events, the basic temporal units in text. These classifications are tailored to the specific task of temporal relation representation. Features closely related to temporal relations are clearly distinguished and represented. For example, whether or not a time expression can be temporally located on the timeline is an important feature about whether or not this time expression participates in temporal relations with other time expressions or events. Therefore, this feature is well represented in our classification for time expressions. And aspect, modality, and eventuality type of an event are important features about how this event interacts with other time expressions and events temporally. Hence, these features are clearly represented in our classification for events as well.

Second, we integrated the concept of reference time from the temporal anaphora theory into our temporal relation representation. Instead of explicitly modeling the temporal relation between every pair of time expressions and events (which models $\binom{n}{2}$ temporal relations for a text with n temporal units), we identify a single reference time for each time expression and event, and only explicitly model the temporal relation between each temporal unit and its reference time (n temporal relations for a text with n temporal units). Temporal relations between other pairs of temporal units are implicitly modeled and can be inferred through the structure of our model.

Third, the integration of reference time enabled us to design a structured representation for temporal relations among basic temporal units in text – the Temporal Dependency Structure. This representation uses the dependency tree structure as the formal object and represent time expres-

sions and events as the nodes on the tree and temporal relations as the edges on the tree. Each parent-child pair on the tree represents a temporal relation between a temporal unit and its reference time.

Finally, we developed detailed annotation guidelines for Temporal Dependency Structure, and annotated a corpus of 235 Chinese documents in two domains: news and narrative – the Temporal Dependency Tree (TDT) corpus. High and stable inter-annotator agreements on this corpus serve as a proof-of-concept for Temporal Dependency Structures, and the corpus also facilitates future research on automatic temporal structure modeling.

Detailed descriptions on design of the Temporal Dependency Structure, our annotation schemes, and statistics on the TDT corpus are presented in Chapter 3.

Automatic Temporal Dependency Structure Parsing

We present the first temporal dependency tree parser in this thesis.

In this research, we developed an end-to-end temporal dependency tree parser. This parser takes a raw text as input, utilizes a neural sequence labeling model to extract events and time expressions, and arrange these events and time expressions in a temporal dependency tree structure based on a neural ranking model.

For comparison, we also developed a strong baseline parser using the logistic regression model and extensive feature engineering, and a few variants of the neural model. These parsers are evaluated on our TDT corpus. Experiments show that both our neural and logistic regression parsers can learn and parse temporal dependency tree structures reasonly well. Considering the observation that different domains (news v.s. narrative) have very different temporal structural patterns, we

show that the neural models hold stronger learning abilities than the logistic regression model and are more adaptive across different domains. Analysis over system output temporal dependency trees are discussed as well.

Detailed descriptions on design of the neural parser and the baseline parsers, our experimental setups, results, and analysis are presented in Chapter 4.

Crowdsourcing Temporal Dependency Structure Annotations

In this thesis, we present a preliminary study on a crowdsourcing approach to efficiently and effectively collect temporal dependency tree annotations.

Since TDT annotation is a very challenging task in a crowdsourcing setup, we designed a crowdsourcing approach which treats the annotation of a complex TDT as two sub-tasks: (1) reference time recognition, and (2) temporal relation identification between every temporal unit and its reference time. Using this approach, we built an English TDT corpus on top of the Timebank. For comparison, we also annotated a subset of this corpus with expert annotators. Annotation experiments show that high Inter-Annotator Agreements can be collected for both subtasks (>80% for IAAs between crowdsoucing workers and experts, and IAAs among crowdsourcing workers). Statistical and linguistic analysis are performed to better understand crowdsourced TDTs, and to compare the differences between crowdsourced and expert-annotated TDTs. We also experimented with our temporal dependency tree parsers on this corpus, achieving comparable results to parsers trained on expert-annotated corpora. Another experiment comparing parsers trained on gold TDTs and parsers trained on crowdsourced TDTs shows that crowdsourcing is an effective approach to collect TDT data. Issues with this crowdsourcing approach are discussed as well, and although the results of this study is still preliminary, it shows promising directions for future research.

Detailed descriptions on design of the crowdsourcing approach, crowdsourcing sub-tasks, corpus analysis, and parsing experiments are presented in Chapter 5.

Chapter 2

Background on Temporal Information Modeling

This chapter will give a background review on prior temporal information modeling research mostly related to our work. The earlier work on temporal information modeling dates back to the 1940s and have gone through a linguistically oriented period when mostly theoretical models on specific temporal phenomena were presented. Important concepts that our work is built upon such as reference time, temporal anaphora, etc. were introduced in the research from this period. We will review these classic work in Section 2.1, with a focus on linguistic theories on temporal anaphora. In the late 1990s and early 2000s, research on temporal information modeling started to focus on data-driven approaches, where corpora of temporal entities (time expressions, events, etc.) and temporal relations were annotated and further automatically predicted. We will review these modern work in Section 2.2, with a focus on their annotation schemes and automatic systems.

2.1 Linguistic Theories on Temporal Anaphora

The notion of *Reference Time* is a long-developed concept. Reichenbach first introduced reference time as part of his conception of *tense* in his influential work *Elements of Symbolic Logic* [Reichenbach, 1947]. Reichenbach claims that there are nine tenses in English: *simple* past, present, and future tense, past, present, and future *perfect* tense, and *posterior* past, present, and future tense (e.g. would, was going to, is going to, will be going to, etc.). Semantically Reichenbach claims that each tense specifies temporal relations among exactly three times particular to a tensed clause/event: the event time (ET), the reference time (RT), and the speech time (ST). More intuitively, consider the example below:

(2): A. John went over to Mary's house.

B. On the way, he had stopped by the flower shop for some roses.

Since the event "went" happened before the speech time, we have $ET_1 = RT_1 < ST$; and the event "stopped" happened before "went", taking "went" as its reference time, we have $ET_2 < RT_2 < ST$ and $RT_1 = RT_2$.

Building upon Reichenbach's conception of reference time, several researchers studied the anaphoric nature of tense in analogy to definite NP and pronoun anaphora. [McCawley, 1971] first explicitly discussed that tense is anaphoric like a definite pronoun. They proposed that the event described in one clause serves as the antecedent of the event described in the next, but that it may be related to the event by being either at the same time or "shortly after" it. [Partee, 1973] also discussed the similarities between tense and definite pronouns in detail, and further discussed anaphoric difference between tense and pronouns in [Partee, 1984]. [Steedman, 1982], [Hinrichs, 1986], and others also argued that Reichenbach' s conception of reference time (RT) is anaphoric. In [Hinrichs, 1986],

Hinrichs makes the simplifying assumption that in a sequence of simple past sentences, the temporal order of events described cannot contradict the order they occur in the text, and focuses on whether the second event follows the previous one or overlaps it. Hinrichs takes advantage of the Aktionsart of a tensed clause, i.e. its Vendlerian classification as accomplishment, achievement, activity, or state (including progressives), and proposes that given a sequence of two accomplishments or achievements, the second event follows the first one, and given a sequence with at least one activity or state, the two events will be interpreted as overlapping each other. Furthermore, [Webber, 1987] discussed examples where tense behaves differently than pronouns anaphorically and proposed that tense is better viewed by analogy with definite NPs rather than with pronouns. [Webber, 1987] also proposed the theory that when processing a narrative text, a listener is building up a representation of the speaker's view of the events and situations being described and of their relationship to one another. This representation was denoted as an *event/situation structure (e/s structure)*. Webber viewed tense and relative temporal adverbials as specifying positions in an evolving e/s structure, and the particular positions they can specify depend on the current context, and the current context only makes a few positions accessible. However, there may be more than one position in the e/s structure which tense can specify and which the new event or situation can attach to. Moreover, [Webber, 1987] introduced *Temporal Focus (TF)* that grounds the context-dependency of tense: At any point N in the discourse, there is always one node in the e/s structure that provides a context for the interpretation of the reference time of the next clause/event, and this node is the temporal focus of the discourse. To track the movement of temporal focus through the progress of a discourse, [Webber, 1987] further proposed four heuristics to manage the temporal focus of a discourse: one *Focus Maintenance Heuristic* to keep the current temporal focus, two *Embedded Discourse Heuristics* to switch the current temporal focus to the reference time of an embedded clause/event, and one *Focus Resumption Heuristic* to return to an earlier temporal focus.

More intuitively, consider the example below:

(3): A. I was$_1$ at Mary's house yesterday.

B. We talked about her brother.

C. He spent 5 weeks in Alaska with two friends.

D. They made a successful assault on Denali.

E. Mary was$_5$ very proud of him.

Event "talked" in sentence B kept the temporal focus of event "was$_1$" in sentence A, while event "spent" in sentence C switched to a new temporal focus, and event "was$_5$" in sentence E returned to the earlier temporal focus as "talked" and "was$_1$".

[Webber, 1987] also pointed out that not only tense can be interpreted anaphorically, temporal adverbs should also behave in a similar manner and are anaphoric too. In a later work, [Webber, 1988] refined her theory on temporal focus and focus management heursitics and proposed the notion *Discourse Anaphors* as expressions with the following two properties: (1) they specify entities in an evolving model of the discourse that the listener is constructing; and (2) the particular entity specified depends on another entity in that part of the evolving "discourse model" that the listener is currently attending to. She discussed examples and how definite pronouns, NPs, and tense share these two properties.

There was also a few work on implementing rule-based systems to distinguish temporal relations between an event and its reference time. [Hitzeman et al., 1995] follows the research line which assumes that by default an event will occur just after a preceding event, while a state will overlap with a preceding event [Kamp, 1979, Hinrichs, 1981, Partee, 1984], and considers exceptions when there is a rhetorical relationship between the two events such as causation, elaboration, or enablement, and the temporal defaults can be overridden, resulting in many possible temporal

relations between two consecutive events. [Hitzeman et al., 1995] proposed a set of constraints that can be used to reduce ambiguities when identifying the temporal relation. These constraints include tense of the two events, cue words such as "because", time expressions, aspects of the two events, and temporal centering.

2.2 Computational Approaches on Temporal Information Modeling

2.2.1 Computational Temporal Modeling Specifications – TimeML

Pre-TimeML Research on Temporal Information Modeling

The Message Understanding Conferences (MUC) included limited annotation of time expressions and temporal information about events. The named entity subtasks of MUC-6 and MUC-7 required the identification of absolute (MUC6) and relative (MUC7) time expressions, however none of these tasks places events on a timeline or relates events temporally to each other. Follow-up work developed a thorough set of guidelines for annotating time expressions [Ferro et al., 2001]. [Setzer, 2002] developed an annotation guideline for time expressions, events, and temporal relations, and annotated a small amount of data in a pilot study. These early guidelines were combined and further developed into the TimeML specification.

TimeML

TimeML [Pustejovsky et al., 2003a] is the most widely used specification markup language for events, time expressions, and temporal relations in natural language text. TimeML has evolved through a few versions. Now publicly available are version 1.1, 1.2, and 1.2.1. Some changes have been made through these versions. For example, some attributes are added/removed/changed on certain annotation objects, and some new relation types or values are added to certain attributes. However, the basics of the TimeML framework stays the same, and in this section, we will give a brief introduction on TimeML and discuss similarities and differences between TimeML and our proposed Temporal Dependency Structure scheme.

In TimeML, four major data structures are modeled: EVENT, TIMEX3, SIGNAL, and LINK. EVENT models situations that *happen* or *occur*. An Event can be punctual or last for a period of time, and includes predicates describing *states* or *circumstances* in which something obtains or holds true. Every EVENT is annotated with its grammatical tense (past, present, future, none) and aspect (progressive, perfective, progressive_perfective, none). One event type out of a set of eight pre-defined types (Occurrence, State, Reporting, Intentional-State, Intentional-Action, Aspectual, Perception, Modal) is also annotated. The TIMEX3 tagset is used to annotate explicit temporal expressions. It is an extension on the TIMEX [Setzer, 2002] and TIMEX2 [Ferro et al., 2001] tagset. Three types of temporal expressions are annotated: Date, Time, and Duration. TIMEX3 also distinguishes temporal expressions based on the level of specification they represent. Some Fully Specified Temporal Expressions are: June 11, 1989; Summer, 2002; etc. And some Underspecified Temporal Expressions are: Monday, next month, last year, two days ago, etc. Examples for Durations are: three months, two years, etc. The function of temporal expressions in the document (such as creation_time, publication_time, etc.) is also annotated on each TIMEX3. Similar to our

Temporal Dependency Structure model, a TIMEX3 has an attribute "anchor time" and an attribute "anchor event" that can be annotated by annotators, and a temporal relation can be annotated between the anchor time/event and this TIMEX3. Utilizing Temporal Functions, TIMEX3 allows delayed computation of the actual value of the temporal expressions in a document. TimeML also annotates SIGNALs. A SIGNAL is usually a section of text (typically function words) that indicate how temporal objects are related to each other. For example, temporal prepositions (e.g. on, during), temporal connectives (e.g. when), subordinators (e.g. if), polarity indicators (e.g. not, no, none), temporal quantifications (e.g. twice, three times) are annotated as SIGNALs.

The LINK tagset in TimeML is used to annotated various relations between the temporal elements in a document, and the temporal orderings between the events in a document. Three types of LINKs are annotated: Temporal Link (TLINK), Aspectual Link (ALINK), and Subordination Link (SLINK). Temporal Links are annotated between pairs of events or pairs of one event and one time. Following [Allen, 1984], TimeML annotates a fine-grained set of temporal relations: before, after, includes, is_included, holds, simultaneous, immediate_after, immediate_before, identity, begins, ends, begun_by, and ended_by. Subordination Links are annotated on pairs of events. The following types of subordination relations are modeled: Modal, Factive, Counterfactive, Evidential, Negative Evidential, and Negative. Aspectual Links are annotated between an aspectual event and its argument event. The aspectual relations modeled in this scheme are: Initiation, Culmination, Termination, and Continuation.

TimeML is the first and most comprehensive temporal information markup language that (1) systematically identifies events and anchors them in time, (a.k.a. time-stamping of events); (2) orders events in text with respect to one another, both intrasentential and intersentential; (3) reasons with under-specified temporal expressions, and allows for delayed interpretations of them; and (4) reasons about the persistence of events, (i.e. how long an event or the outcome of an event lasts).

2.2.2 Pair-wise Temporal Relation Modeling

2.2.2.1 Pair-wise Temporal Relation Schemes and Corpora

Timebank, AQUAINT_TimeML

Most pair-wise temporal relation annotation schemes are rooted from the TimeML specifications. Many corpora are annotated based on TimeML and automatic systems are developed and trained on these datasets. The TimeBank 1.1 corpus [Pustejovsky et al., 2003b] is an illustration and proof of concept of the TimeML specifications. It was created in the early days of TimeML and follows the 1.1 version of the specifications. The text sources for Timebank 1.1 is from a wide variety of media sources in the news domain. It contains texts from the Document Understanding Conference (DUC) corpus, Automatic Content Extraction (ACE) program texts, and the Penn Treebank texts (i.e. Wall Street Journal newswire texts). It was annotated partly by experts and partly by non-experts who were trained first and their annotations were reviewed by experts afterwards. The annotation process consists of a preprocessing stage followed by an actual human annotation stage. Both stages utilize many automation tools. The preprocessing stage does automatic temporal expression recognition, automatic event recognition, and automatic labeling of event tense, aspects, etc. During the human annotation stage, the results from the preprocessing stage are manually checked and full annotations are added. Automatic temporal closure and graphic visualization are performed to assist human annotation. Timebank 1.1 consists of 300 news documents in total (68,555 words). The basic corpus statistics for Timebank 1.1 are presented in Table 2.1 and Table 2.2.

The Timebank 1.2 Corpus[1] follows the newer TimeML specifications version 1.2.1. The annotation

[1] http://www.timeml.org/timebank/documentation-1.2.html

Tag	Count
Event	7,571
Timex	1,423
Signal	2,212
Total	11,206

Table 2.1: Number of events, timex, and signals in Timebank 1.1.

Link Type	Count	%
Tlink	5,132	62.2
Slink	2,857	34.7
Alink	253	3.1
Total	8,242	100

Table 2.2: Distribution of TLINK, SLINK, and ALINK in Timebank 1.1.

was performed on news articles from Automatic Content Extraction (ACE) program texts and the Penn Treebank2 Wall Street Journal texts. The annotation process for Timebank 1.2 is similar to Timebank 1.1, except that all annotations are performed by expert annotators for this version. Timebank 1.2 contains 183 news articles in total (61,000 words). The basic corpus statistics for Timebank 1.2 are illustrated in Table 2.3 and Table 2.4.

Tag	Count
Event	7,935
Timex	1,414
Signal	688
Total	10,038

Table 2.3: Number of events, timex, and signals in Timebank 1.2.

Inter-Annotator Agreements are computed on two expert annotations on a subset of ten documents in Timebank 1.2. Exact match F-score is used as the IAA metric. Table 2.5 shows the IAA scores for each annotation object.

The AQUAINT_TimeML corpus is another corpus annotated with TimeML scheme 1.2.1. It consists of news reports from four topics in the novelty track of the Text REtrival Conference (TREC)

Link Type	Count	%
Tlink	6,418	66.7
Slink	2,932	30.5
Alink	265	2.8
Total	9,615	100

Table 2.4: Distribution of TLINK, SLINK, and ALINK in Timebank 1.2.

Annotation Type	F
Timex3	.83
Event	.78
Signal	.77
Tlink	.55
Slink	.85
Alink	.81

Table 2.5: IAAs of Timebank 1.2 annotations.

2003 and 2004[2]. The four topics are: Kenya Tanzania Embassy bombings; Elian Gonzalez Cuba; NATO, Poland, Czech Republic, Hungary; and, Slepian abortion murder. These particular sources were chosen because they offered text rich with temporal information both in the form of temporal expressions and events that could be anchored or ordered in time. AQUAINT_TimeML contains 73 news reports in total (35,000 words). Basic statistics of this corpus are shown in Table 2.6 and Table 2.7. However, this corpus was annotated with single expert annotations only, and no IAAs are reported.

Tag	Count
Event	4,432
Timex	605
Signal	268
Total	5,305
Makeinstance	4,432

Table 2.6: Number of events, timex, and signals in AQUAINT_TimeML.

[2]The novelty track of TREC 2003 and 2004 use documents from the AQUAINT collection: `https://catalog.ldc.upenn.edu/LDC2002T31` (Advanced Question-Answering for Intelligence), collected by researchers at UPenn.

Link Type	Count	%
Tlink	5,365	87.8
Slink	675	11.0
Alink	71	1.2
Total	6,111	100

Table 2.7: Distribution of TLINK, SLINK, and ALINK in AQUAINT_TimeML.

TempEval Tasks

The TempEval tasks are a series of SemEval shared tasks aimed at automatic temporal information modeling, including time expression and event recognition, and temporal relation identification. Through these tasks, a number of automatic temporal relation systems were developed. We will give a brief description of the tasks in this section, and overview their participating systems and other automatic temporal systems in the following sections.

Through the course of a decade (2007∼2017), six TempEval tasks were held in total. The first three of them focus on the domain of news reports [Verhagen et al., 2007a, Verhagen et al., 2010a, Verhagen et al., 2010a], and the second half focus on the clinical domain [Bethard et al., 2015a, Bethard et al., 2016a, Bethard et al., 2017]. The first TempEval task [Verhagen et al., 2007a] was held in 2007. This was the first time temporal information processing was evaluated in a shared task setup. To implement a straightforward evaluation, they broke down the full task of temporal information processing into three smaller subtasks that allow pairwise evaluation of temporal relations. The three subtasks are: (A) for each event, classify the temporal relation between it and all timex in the sentence; (B) for each event, classify the temporal relation between it and the Document Creation Time (DCT); and (C) for each pair of main events from two consecutive sentences, classify the temporal relation between them. The annotation scheme used in this task is a simplified version of TimeML. Namely, a subset of TimeML tag set (TIMEX3, EVENT, and TLINK) was used, and a simplified temporal relation set was implemented (only six temporal relations are distinguished

in TempEval-1: Before, After, Overlap, Before-or-Overlap, Overlap-or-After, Vague). Modified Timebank1.2 (using the simplified TimeML scheme) served as the training data, and newly annotated news data was used as the test set. TempEval-1 initiated a years-long effort in temporal information processing research and contributed to very straightforward and manageable temporal evaluations. Some limitations are that this task focused on temporal relation identification and didn't include time expression and event identification as part of the evaluation. Moreover, only a subset of events (events whose stem occurs 20 times or more in Timebank) are included in this task.

Some of the limitations of TempEval-1 were addressed in TempEval-2 [Verhagen et al., 2010a]. Although the same simplified TimeML scheme was applied for data annotation for TempEval-2, this second TempEval task covered subtasks on time expression and event identification, including time expression extraction, classification, and normalization, and event extraction and classification. It also extended the number of evaluations on temporal relation identification, including all three subtasks from TempEval-1, and the temporal relation between two events in a sentence where one syntactically dominates the other. Another major contribution of TempEval-2 is that it extended the task from English-only to six different languages: English, Spanish, French, Italian, Chinese, and Korean. Although the final participating systems only focused on two languages (English and Spanish), the multilingual temporal annotations collected in this task supported a number of future research efforts.

Both TempEval-1 and TempEval-2 utilized gold Timebank1.2 as their main training data (and a small number of newly annotated news articles as the test data). TempEval-3 [UzZaman et al., 2012] was the first time the gold AQUAINT temporal corpus, and a large automatically system-annotated temporal corpus were added as parts of the training data. The inclusion of these new corpora resulted in a training dataset that is ten times bigger, yet only a small portion of it was gold

standard. The system-annotated "silver" temporal corpus was generated by applying then state-of-the-art temporal systems [Llorens et al., 2010, Llorens et al., 2013, UzZaman and Allen, 2010] on Gigaword [Parker et al., 2011]. Participating systems' results show that this "silver" training data doesn't help timex extraction or temporal relation classification, but is useful for event extraction. TempEval-3 also provided a "platinum" test dataset, which has higher Inter-Annotator Agreement (IAA) scores than previous test sets and existing TimeML corpora. Another major difference in TempEval-3 was that the full set of TimeML temporal relations was used (instead of the simplified six relations). And end-to-end temporal relation identification was for the first time evaluated, where participants were given raw texts and need to perform timex, event identification together with temporal relation identification. TempEval-3 also included both English and Spanish, using the revised and finalized Spanish Timebank1.0 as training and test data for Spanish.

While all data annotated and used in the first three TempEval tasks are within the domain of news reports, the following TempEval tasks focused on data in the clinical domain. Clinical TempEval 2015 [Bethard et al., 2015a] utilized a modified/extended version of TimeML developed by the THYME project [Styler IV et al., 2014a, Styler IV et al., 2014b]. The extensions were specialized for the clinical domain, such as new timex types for words indicating particular clinical temporal locations. For example, in the following clinical notes, "postoperative" is a TIMEX3 of type PrePostExp, indicating the temporal location after the "operation" event.

(4): The patient did not have any postoperative bleeding.

New event attributes were also added to represent special features of clinical events. For example, for event "slight nausea", the DEGREE attribute of this event should be LITTLE. One limitation of this modified TimeML scheme is that it only models two types of temporal relations: the "contain" TLINKs, and the temporal relations between events and the DCT. That is, temporal relations

between events such as "before", "after", and "overlap" are not modeled under this scheme. The annotation of this clinical temporal corpus was carried out through the THYME project [Styler IV et al., 2014a, Styler IV et al., 2014b]. Their data source was clinical notes and pathology reports from colon cancer patients at Mayo Clinic. The corpus contains 293 documents for training, and 147 documents for test. Nine subtasks were evaluated in total, including timex extraction and classification, event extraction and labeling, and temporal relation identification between events and the DCT, and between events and/or timex with the "contain" TLINKs. They also performed evaluations on both end-to-end setup, and temporal relation only setup with gold timex and event available.

Based on the first clinical TempEval, Clinical TempEval 2016 [Bethard et al., 2016a] added more data (151 documents) for participants, included more participating systems, and reported major performance improvements both on timex/event recognition and temporal relation identification, although the latter remained a challenging problem.

Clinical TempEval 2017 aimed at answering the question: how well can temporal systems trained on one medical condition perform on a different medical condition. In other words, this TempEval task inquires how domain adaptation techniques can be applied to analyze temporal information on a new medical condition that doesn't have much annotated training data. Original data on colon cancer patients was provided as the main domain, and new clinical records from brain cancer patients were added as the new domain. Evaluation setups include (1) both training and testing on the main domain, (2) training on the main domain and testing on the new domain, and (3) training on both domains and testing on the new domain. Since temporal information modeling is already a quite complicated task, only a few domain adaptation techniques were applied in participating systems, and their results show that developing temporal systems that work across different medical conditions was still a very challenging task.

Table 2.8 summarizes the basic information for the six TempEval tasks.

Year	Domain	Language	Data	Tasks
2007	news	English	modified Timebank1.2, new gold data for test	3 tasks: temp rel only
2010	news	6 languages	English: modified Timebank1.2, Spanish: Spanish Timebank	6 tasks: timex, event, & temp rel
2013	news	English & Spanish	English: modified Timebank1.2, AQUAINT, new auto silver data, new platinum data for test; Spanish: Spanish Timebank1.0	5 tasks: timex, event, temp rel, & end-to-end
2015	clinical	English	THYME corpus (train + dev) (clinical records of colon cancer patients from Mayo Clinic)	9 tasks: timex, event, temp rel, & end-to-end
2016	clinical	English	THYME corpus (train + dev + test) (colon cancer records)	(same 9 tasks as above)
2017	clinical	English	THYME corpus (colon + brain cancer records)	(same 9 tasks as above)

Table 2.8: Six TempEval tasks summaries.

2.2.2.2 Comparisons between Pair-wise Models and TDT

As mentioned before, Timebank annotation guidelines specify temporal relation annotations between events and times in a pair-wise fashion. Pairs of events and time expressions in a document are considered and the ones that annotators regard as having temporal relations are annotated with a TLINK. Therefore, theoretically a maximum of $\binom{n}{2}$ possible temporal relations could be annotated for a document with n events and temporal expressions, however, practically annotators would

pick much less pairs. In contrast, we propose to model temporal relations in a document in a more structured way. For each document, we build a dependency tree structure that represents events and times as nodes and temporal relations among them as edges. This is the main difference between pair-wise models and our proposed temporal relation representation model. Our structured model has the advantage of lower annotation complexity. The TIMEX3 attributes "anchor time/event" in TimeML is very similar to our parent-child structures in a temporal dependency tree. However, TimeML only models these anchors for TIMEX3s, not EVENTs, and their annotation is optional.

Another difference between TimeML and our scheme is our different treatment to temporal expressions. TimeML annotates explicit temporal expressions, including expressions that are not actually temporal locations (i.e. not anchorable on a timeline). For example, in the following sentence A, "three months" is an explicit temporal expression and should be annotated as a Duration in TimeML. However, this expression is not describing a temporal location on a timeline. In our design, we focus more on the time-stamping of events and/or times. However, non-temporal location time expressions are not helpful for anchoring events and/or times (e.g. the example sentence A below). Some examples of temporal location time expressions are as follows. The temporal location of the time expression "3 days after New Years Eve" in sentence B is January 3rd on the year of the Document Creation Time. And the temporal location of the second time expression "10 minutes later" in sentence C is 8:10am on the date of the DCT.

(5): A. This procedure usually takes three months.

B. He left 3 days after New Years Eve.

C. He arrived at 8:00am. 10 minutes later, the class began.

Moreover, among temporal location time expressions, we propose a more detailed categorization of temporal expressions. Instead of recognizing fully specified temporal expressions from un-

derspecified ones, we distinguish if a temporal expression is "absolute" or "relative", and if it is "concrete" or "vague" (see Chapter 3 Section 3.2 for detailed explanations and examples). These two distinctions not only make temporal expressions form consistent tree structures in our proposed temporal dependency trees, and also make downstream temporal expression normalization easier.

Additionally, to make our first stage annotation experiments efficient, we also simplified over TimeML on certain aspects. For example, we don't annotate SLINK, ALINK, and SIGNALS; we apply less annotations on temporal expressions and events (instead of annotating tense, aspect, and event type on every event, we only annotate one event type from a specially designed set that covers these three aspects); we use a simplified temporal relation set and don't explicitly represent magnitudes; and we are not explicitly modeling event coreference yet.

2.2.2.3 Automatic Pair-wise Temporal Relation Identification Systems

Rule-based and Statistical Machine Learning Systems

The TempEval shared tasks have inspired a large number of research efforts on automatic computational modeling for temporal relation and information. Some of the early temporal systems focused on rule-based approaches. [Hagège and Tannier, 2007] utilized a rule-based deep syntactic analyzer for temporal expression identification, and a rule-based linguistic analyzer for temporal relation identification. [Strötgen and Gertz, 2010] built the system HeidelTime, one of the state-of-the-art systems for time expression extraction and normalization. It is a rule-based system mainly using regular expression patterns for the extraction of time expressions as well as knowledge resources and linguistic clues for their normalization. The later improved version of HeidelTime was extended to 13 languages with hand-crafted resources, and even more languages with automati-

cally created resources [Strötgen et al., 2013, Strötgen et al., 2014, Li et al., 2014, Manfredi et al., 2014, Strötgen and Gertz, 2015]. [Saquete, 2010] deployed a rule-based system using knowledge databases for time expression identification. [Chang and Manning, 2013] built a rule-based time expression tagger based on regular expression patterns over tokens. [Zavarella and Tanev, 2013] utilized finite-state rule cascades to recognize and classify time expressions and events. [Tissot et al., 2015] built its in-house rule-based systems for clinical temporal modeling.

Many systems integrated rule-based components together with statistical models. [Min et al., 2007], on one hand, took advantage of a syntactic pattern matching tool and deployed hand-crafted finite state rules for temporal expression labeling and normalization, and utilized heuristics, a lexicon, and lexical features such as lemmas, parts of speech, and WordNet senses for event detection. On the other hand, they also engineered various syntactic and semantic features for its statistical models for temporal relation identification. [Puşcaşu, 2007] implemented a rule-based temporal reasoning mechanism for intra-sentence temporal relations. It leveraged the process of sentence-level syntactic tree generation to perform bottom-up propagation of temporal relations between syntactic constituents. Heuristics were used for temporal conflict resolution. Inter-sentence temporal relation identification, however, were identified with both heuristics and statistical models. [Vicente-Díez et al., 2010] used rules plus simple statistics to tackle time expression extraction, classification, and normalization in Spanish text. [Grover et al., 2010] utilized a rule-based syntactic analyzer, and experimented with both rule-based and logistic regression models for time and event identification. [Kolya et al., 2010] built rule-based systems for time and event identification and employed CRF models for temporal relation identification. [Kolomiyets and Moens, 2010] focused on time expression and used a logistic regression model for extraction and a rule-based system for normalization. To extend positive annotations in the corpus, they also exploited semantically similar words automatically obtained from a large un-annotated textual corpus. [Derczynski

and Gaizauskas, 2010] employed a rule-based system for time expression identification, and a logistic regression classifier for temporal relation identification, using features such as associated temporal signal words. [Chambers, 2013] used both logistic regression classifiers and rule-based systems for the whole pipeline of temporal information modeling. [Filannino et al., 2013] utilized CRF models for time expression extraction and an off-the-shelf rule-based system for their normalization. [Cohan et al., 2016] built CRF and logistic regression models using lexical, morphological, syntactic, dependency, and clinical domain specific features, combined with pattern matching rules. [Sarath et al., 2016] experimented with an ensemble of rule-based and statistical model using lexical, syntactic, and morphological features. [Grouin and Moriceau, 2016] incorporated the rule-based system Heidel-Time from [Strötgen et al., 2013] into its CRF models with lexical, morphological, and word cluster features. [Barros et al., 2016] utilized both SVM models with lexical and morphological features and rule-based extensions to Stanford CoreNLP [Manning et al., 2014]. [MacAvaney et al., 2017] built ensembles of CRFs, rules, and decision trees using character n-grams, lexical, word clusters, word embeddings, parts of speech, syntactic, dependency tree paths, semantic role, and UMLS concept types as features. [Lamurias et al., 2017] combined CRFs and rules with character n-grams, words, parts of speech, and UMLS concept types features.

A number of pure statistical temporal systems have been developed as well, with the most commonly used models being SVMs, CRFs, Decision Trees, Structured Perceptrons, and Logistic Regression models. [Bethard and Martin, 2007] trained standard SVM models for temporal relation identification using syntactic features and gold Timebank label features. [Cheng et al., 2007] utilized features from dependency parsing trees and built a sequence labeling model for temporal relation identification. [Hepple et al., 2007] took advantage of the off-the-shelf machine learning suite WEKA and used a classification model with lexical and Timebank label features. [Llorens et al., 2010] implemented the system TIPSem, one of the state-of-the-art systems in temporal

modeling on both English and Spanish. TIPSem utilized CRF models with semantic role features. A comparison experiment between TIPSem with and without semantic features showed that semantic information is very important for time expression identification. TIPSem was also used to generate silver training data for a later clinical TempEval. [UzZaman and Allen, 2010] employed systems that use a combination of deep semantic parsing, Markov Logic Networks, and CRF models to tackle the entire temporal pipeline of time, event, and temporal relation identification. [Ha et al., 2010] built a statistical system using Markov Logic in combination with rich lexical relation features as well as lexical and syntactic features. [Jung and Stent, 2013] aimed at time expression and event identification with logistic regression classifiers, and experimented with various sets of features, including basic lexical features, rich syntactic features, and rich semantic features. [Bethard, 2013] built a pipeline of statistical models, each with a small set of simple morpho-syntactic features, for the complete pipeline of time, event, and temporal relation identification. [Kolya et al., 2013] employed CRF models for each part of the temporal processing pipeline, using various features based on different lexical, syntactic and semantic information, extracted with Stanford CoreNLP and WordNet-based tools. [Kolomiyets and Moens, 2013] used logistic regression models for time expression and event identification, and deployed a shift-reduce temporal dependency parser [Kolomiyets et al., 2012] in a pair-wise temporal relation identification scenario. This is a temporal dependency parser that is comparable to our proposed temporal dependency representation, and is described in greater detail in Section 2.2.3.2. [Laokulrat et al., 2013] built logistic regression classifiers for temporal relation classification and exploited features extracted from a deep syntactic parser, including paths between event words in phrase structure trees and their path lengths, and paths between event words in predicate argument structures and their subgraphs. [Velupillai et al., 2015] participated in the Clinical TempEvals and their supervised classifiers used features generated by the Apache clinical Text Analysis and Knowledge Extraction

System (cTAKES[3]). [Tissot et al., 2015] built supervised classifiers using SVM models. [Chikka, 2016] experimented with CRF and SVM models. [Hansart et al., 2016] built CRF models with lexical features. [Lin et al., 2015] and [Leeuwenberg and Moens, 2016] also utilized the cTAKES with additional feature engineering. [Tourille et al., 2016] experimented with SVM models either using lexical, syntactic, and structural features or using word embeddings with no hand-crafted features. [Abdulsalam et al., 2016] built CRFs and SVMs using lexical, morphological, syntactic, character pattern, character n-gram, and gazetteer features. [Lee et al., 2016] implemented SVMs using lexical, morphological, syntactic, discourse, and word representation features. [Caselli and Morante, 2016] built CRFs with morpho-syntactic, lexical, UMLS[4], and DBpedia[5] features. [Leeuwenberg and Moens, 2017] combined SVMs with structured perceptrons using word and part of speech features, as well as preliminary domain adaptation techniques for data in the clinical domain. [Huang et al., 2017] built and ensemble of SVMs and CRFs with word n-grams, parts of speech, named entities, dependency trees, and UMLS concept types as features.

Neural Systems

Later temporal systems started to utilize neural models more often. [Fries, 2016] implemented recurrent neural networks with word embeddings for end-to-end time expression, event, and temporal relation identification. [Chikka, 2016] and [Li and Huang, 2016] also used neural models for both end-to-end and temporal relation identification only on the clinical domain. [Sarath et al., 2017] built and ensemble of CRFs, rules, neural networks, and decision tree using character n-grams, word n-grams, word embeddings, verb tense, section headers, and sentence embeddings as features. [Tourille et al., 2017] combined recurrent neural networks with character and word

[3] https://ctakes.apache.org
[4] Unified Medical Language System
[5] https://wiki.dbpedia.org/

embeddings and SVMs with lexical and part of speech features, as well as preliminary domain adaptation techniques for the clinical domain. [Long et al., 2017] built an ensemble of rules, SVMs, and recurrent and convolutional neural networks with words, word embeddings, and verb tense as features for temporal information processing in the clinical domain.

A representative work on neural systems for pair-wise temporal relation extraction is described in [Dligach et al., 2017], which empirically shows that CNNs and LSTMs can be successfully used for temporal relation extraction (establishing state-of-the-art results), without manually engineered features (with only word tokens and/or pos tags). [Dligach et al., 2017] claims that the vast majority of systems in temporal information extraction challenges, such as the i2b2 [Sun et al., 2013] and Clinical TempEval tasks [Bethard et al., 2015b, Bethard et al., 2016b], used classifiers with a large number of manually engineered features, which experience a significant accuracy drop when applied to out-of-domain data [Wu et al., 2014, McClosky et al., 2010, Daumé III, 2009, Blitzer et al., 2006]. Therefore, they proposed two neural architectures for temporal relation extraction: a convolutional neural network CNN [LeCun et al., 1998] and a long short term memory neural network LSTM [Hochreiter and Schmidhuber, 1997], which require minimum manual feature engineering. They also proposed a new simple method to mark the positions of the relation arguments: XML tags are used to mark the positions (e.g. <e1> diagnosed </e1>, <t> may of 2010 </t>). This representation of relation argument positions can be used directly by neural models.

More specifically, with a concatenation of n words and/or POS embeddings of dimension d as the input representation, they built separate models for event-time relations and event-event relations. Standard split Clinical TempEval 2016 corpus is used as their experimental data, focusing only on the "contains" relation, and the THYME system [Lin et al., 2015], an SVM classifier with hand-engineered linguistic features which achieved the highest performance on Clinical TempEval 2015 test set [Lin et al., 2016], is used as the baseline. They re-trained two versions of this baseline

system on their own experimental data: one with the full set of features, and one with only word tokens as features. Different sets of experiments are conducted to compare their systems against the baseline systems, on all events or on medical events only.

They discovered that CNN with only word tokens as features is the best performing model among their neural models; for event-time "contains" relations, neural models in general outperform the traditional feature-based baseline model, but for event-event "contains" relations, none of their neural models outperform the baseline; when only considering medical events, their best neural model (CNN with word tokens) outperforms baseline on both event-time relations and event-event relations; and their proposed new simple encoding for relation argument positions outperform the previous encoding method (position embeddings). They also discussed that CNN with only POS tags as features outperforming the feature-based baseline suggests that POS tags alone is enough for this task when coupled with neural models; CNN models outperform LSTM models on this task; and the reasons that their neural models don't perform as well as the baselines might be: event-event relations are much more difficult than event-time relation, and the class imbalance issues. The relation:none-relation ratio for event-event relations is 1:15, and the baseline system is tuned with class specific weights that help it deal with class imbalance.

2.2.3 Temporal Structure Modeling

2.2.3.1 Temporal Structure Schemes and Corpora

Although structured interpretation of temporal relations in discourse is a long-developed concept (as introduced in § 2.1), only a few works have been done on the design of annotation schemes of temporal structures, the actual annotation on data, and the construction of scalable corpora. Here

we give a brief introduction to related work in this area.

Temporal Dependency Structure for Narrative Events

[Bethard et al., 2012] introduced a new temporal relation annotation scheme different from all pair-wise schemes – annotating events in a narrative story as a temporal dependency tree structure. In their scheme, annotators were instructed to link each event in the story to a single nearby event, similar to what has been observed in reading comprehension studies [Johnson-Laird, 1980, Brewer and Lichtenstein, 1982]. When there were several reasonable nearby events to choose from, the annotators were instructed to choose the temporal relation that was easiest to infer from the text (e.g. preferring relations with explicit cue words like *before*).

They experimented with a few different scheme designs. On event annotation, they did three different annotation schemes: (1) TimeML event identification rules; (2) TimeML events without events in direct speech and negated, modal, or hypothetical events; and (3) all events in (2) without light verbs and aspectual verbs. On temporal relation annotation, they did two different annotation schemes on two relation label sets: (1) Before, After, Overlap; and (2) Before, After, Includes, IsIncluded, Identity, Overlap. 20 stories are annotated with these different annotation schemes. And using Krippendorff's nominal Alpha [Krippendorff, 2004, Hayes and Krippendorff, 2007] as the inter-annotation agreement (IAA) measure, their experimental annotations show that event recognition scheme (3) obtains the highest IAA, and relation label set (1) gets higher IAA than relation set (2). They then performed annotation on 100 fables using the best event scheme (3) and the more detailed relation label set (2). And 0.856, 0.822, and 0.700 IAAs are reported respectively for event recognition, event links, and event ordering relation labels.

[Bethard et al., 2012] shows that temporal relations between events in narrative stories can be

accurately annotated as a form of temporal dependency structure, where all events in the plot are connected by a single spanning tree. Additionally, pointing out that the problem of prior work on temporal relation annotation is that they generate disconnected timelines, [Bethard et al., 2012] claims that their annotation scheme guarantees a connected dependency tree, therefore connected timelines.

However, [Bethard et al., 2012] only did annotations on narrative stories, and only worked with events, while we propose a more complete scheme on different types of text (both narrative stories and reporting news), and cover both events and time expressions in the temporal dependency tree structure. [Bethard et al., 2012] requires annotators to link each event in a story to a single nearby event with which the temporal relation is the easiest to infer, using the annotators' own judgment, while we require annotators to find the reference time for each event and time expression, using the well-developed linguistic concept of temporal anaphora.

Multi-axis Annotation Scheme for Event Temporal Relations

Another structured representation of temporal relations among events is presented in [Ning et al., 2018b]. They proposed a multi-axis annotation scheme to capture temporal structures among events. Under the observation that not all pairs of events should be annotated a temporal relation, [Ning et al., 2018b] proposed to model events on different axes, and only events on the same axis should be considered temporally. More specifically, they proposed to have eventive events on the main axis, INTENTION and OPINION events on an orthogonal axis, HYPOTHESIS and GENERIC events on a parallel axis, NEGATION events not on any axis, and STATIC and RECURRENT events on the OTHER axis. Their annotation process is as follows: (1) classify if an event is anchorable to a given axis (i.e. an event type classification step); (2) annotate every

pair of events on a given axis; (3) repeat on every axis. This model allows annotators to focus on only comparable pairs, avoiding situations where annotators are forced to relate event pairs that have none or very vague temporal relations. "Orthogonal Axes" is a novel design in this scheme. Intersection events of two orthogonal axes can be compared to events on both axes, and can sometimes bridge events, especially for INTENTIONS and OPINION events. They also observed that in previous annotation work, event end-points are a major source of annotation disagreements, and proposed annotation on start-points only.

A pilot expert annotation experiment shows a great improvement in IAA (.6 to .8 Cohen's Kappa) when using their annotation scheme. They used crowdsourcing to annotate the entire Timebank-Dense with their annotation scheme, and showed good ACC (accuracy compared to expert annotation) and WAWA (Worker Agreement with Aggregate) scores. A comparison on the 1.8k event pairs that their annotation and the original Timebank-Dense annotation have in common shows that: the two annotations have a high agreement level, and due to the interval-splitting technique, their annotation has more specific temporal relation labels for the "vague" relations in Timebank-Dense.

They also trained two baseline temporal relation recognition systems (an averaged perceptron system). One on their annotations and the other on the original Timebank-Dense annotations. They reported better results on their annotations over the original annotations.

[Ning et al., 2018b] focuses only on temporal relations among events, excluding time expressions which play an important role in inferring temporal relations. [Zhang and Xue, 2018b] includes both time expressions and events in their temporal structure. [Ning et al., 2018b] models temporal relations between events on one axis pair-wisely, not capturing the structural relations among them. Additionally, "vague" temporal relations (which are usually annotated between events on

different axes) are left un-attended in their scheme. They define the process of "axis projection" to be projecting events across different axes, and through this projection, figuring out the temporal relations between them. However, due to difficulties in this projection, they focus only on same-axis relations in the current stage. This leaves temporal relations across different axes un-interpreted. [Zhang and Xue, 2018b] models the temporal relations in a text in one simple structure using which temporal relations between every pair of time expressions and/or events can be inferred.

Temporal Discourse Models (TDM) for Narrative Events

[Mani and Pustejovsky, 2004] proposed the Temporal Discourse Models (TDM) – tree-structured representations for the temporal structures of narratives. Nodes in the tree represents abstract events (interpreted as pairs of time points), and the temporal relations represented in the tree are "temporal inclusion" relations and temporal ordering relations are represetned as a separated set of constraints. More specifically, a TDM is a pair $\{T, C\}$, where:

- T is a rooted, unordered, directed tree with nodes $N = \{E \cup A\}$, where a pair of parent-child represents a "temporal inclusion" relation.
    - E is the set of events mentioned in the text.
    - A is a set of abstract events.
- C is a set of temporal ordering constraints, using the ordering relations: $<$ and $\subseteq_{min}$.
    - $<$ represents temporal precedence.
    - $\subseteq_{min}$ represents minimal inclusion (for *States* only, see below for further explanation).

Consider the following example:

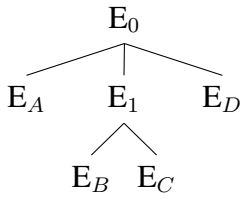
(6): A. John went into the florist shop.

B. He had promised Mary some flowers.

C. She said she wouldn't forgive him if he forgot.

D. So he picked out three red roses.

The TDM tree for this discourse is:



And the TDM constraints for this discourse are: $C : \{E_B < E_C, E_C < E_A, E_A < E_D\}$, where E_0 and E_1 are abstract events, and $E_A \sim E_D$ are events corresponding to sentence $A \sim D$.

[Mani and Pustejovsky, 2004] focuses only on temporal relations among events in narratives, while our proposed representation includes both time expressions and events in their temporal structure, and supports both narrative stories and news reports. TDMs represents "temporal inclusion" relations in its tree structure, and temporal precedence relations in a separated set of constraints. This approach models temporal relations in two different representations, which could potentially add difficulties for automatic systems, and doesn't explicitly model the "overlap" temporal relation. We propose to include all of the basic temporal relations ("before", "after", "overlap", and "includes") together in one consistent tree structure.

[Mani and Pustejovsky, 2004] proposed a special treatment for stative events (i.e. *States*), whose temporal relation with other *States* or *Events* are often not very explicitly stated. For example, in the following discourse, it can be inferred with certainty that at the same time that E_A happened, E_B was true. However, it is possible that the *State* E_B extends before and/or after E_A. TDM

chooses not to capture such information, and only represents that E_B is "minimally" (i.e. "at least") included in E_A. The TDM tree and constraints for this example are shown below.

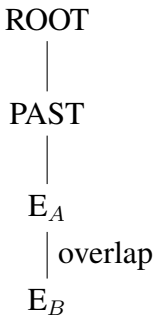
(7): A. John walked home.

B. He was feeling great.

TDM tree:

TDM Constraints: $C : \{E_B \subseteq_{min} E_A\}$.

This approach of handling stative events is similar to our temporal structure design. However, we utilize the "overlap" relation and tend not to model a temporal inclusion relation if it is unclear in the context, and all information is represented on the tree structure. For the above example, our Temporal Dependency Tree (TDT) structure would be as follows (see Chapter 3 for more introduction on TDTs).



Narrative Containers

Narrative Containers is another structured representation for temporal information in text. [Pustejovsky and Stubbs, 2011] first proposed the notions of narrative containers and narrative times. In short, the narrative container of a document is an assumed time window before DCT of the document. And the size of the window is dependent on the genre of the article. For example, a daily news papers' narrative container is one day leading up to the release time of the news paper; newswire articles' narrative containers are maybe about 2∼10 hours; and monthly journals' narrative containers are roughly one month, etc. The narrative time of a document is the current temporal anchor for events in the text, set by a time expression or an event; and it can change as the reader moves through the narrative. For example, the first snippet below is from an article published on the Wall Street Journal. Since Wall Street Journal is a daily newspaper, its assumed narrative container size is one day. Therefore, the event "adopted(e1)" is highly likely to have happened on 10-25-1989, the 1-day time period leading up to the DCT of this document. An example for narrative times is presented in the second text. Three time expressions function as narrative times: DCT (t0), Sunday (t1), and earlier this month (t2). As the reader goes through the text, its narrative time shifts and events are contained by according narrative times (e1, e4 are in the container t1; e2, e3, e5, and e6 are in the container t2).

(8): A. DCT: 10-26-1989$_{t0}$

1 Philip Morris Cos., New York, adopted$_{e1}$ a defense measure designed to make a hostile takeover prohibitively expensive.

B. DCT: April 25, 2010 7:04 p.m. EDT$_{t0}$

President Obama paid$_{e1}$ tribute Sunday$_{t1}$ to 29 workers killed$_{e2}$ in an explosion$_{e3}$ at a West

> Virginia coal mine earlier this month$_{t2}$, saying$_{e4}$ they died$_{e5}$ "in pursuit of the American dream." The blast$_{e6}$ at the Upper Big Branch Mine was the worst U.S. mine disaster in nearly 40 years.

The Narrative Containers Model was later extended and adapted to the clinical domain in the THYME project [Miller et al., 2013]. THYME annotation guidelines recognizes four narrative containers regarding the DCT of a document: before DCT, overlap DCT, before and overlap DCT, and after DCT. It also allows time expressions and events in the text to function as narrative containers (merging the notion narrative container and narrative time). And lastly, on top of the temporal container relations, temporal ordering relations are modeled as well.

2.2.3.2 Automatic Temporal Structure Parsing Systems

[Kolomiyets et al., 2012] is one of the very few work on automatic temporal structure parsing. Based on the temporal dependency structure introduced by [Bethard et al., 2012], they built two temporal dependency parsers using traditional dependency parsing techniques. These parsers are trained and evaluated on the corpus developed in [Bethard et al., 2012], 100 fable stories. (See Section2.2.3.1 for more descriptions on this dependency structure and corpus.)

More specifically, they built a Shift-Reduce Parser (SRP), using the Covington set of transitions [Covington, 2001] as it allows for parsing non-projective trees; and a Graph-Based Parser, using Maximum Spanning Tree (MST) [Georgiadis, 2003] with the Chu-Liu-Edmonds algorithm [Chu and Liu, 1965, Edmonds, 1967], with Margin Infused Relaxed Algorithm (MIRA) [Crammer and Singer, 2003, Crammer et al., 2006] for predicting edge scores. These two parsers are evaluated against two baseline systems: LinearSeq, linking all events in linear order with BEFORE relation; and ClassifySeq, linking all events in linear order, with a trained pair-wise classifier to predict the

relation. And they used Unlabeled/Labeled Attachment Score (UAS/LAS) and Unlabeled/Labeled Tree Edit Distance Scores (UTEDS/LTEDS) as the evaluation metrics.

Their experimental results show that the two traditional parsers both match or beat the two baselines on unlabeled evalutations, and both outperform the two baselines on labeled evaluations. They discovered that ClassifySeq works basically identical with LinearSeq, showing that simple pair-wise classifier was unable to learn anything beyond predicting all relations as BEFORE. They also showed that SRP performs better than MST on labeled evaluation, likely because SRP allows for features describing previous parse decisions.

[Kolomiyets et al., 2012] built two temporal structure parsers based on the temporal dependency structures introduced in [Bethard et al., 2012], which captures only the temporal relations among events. Our parsers, on the other hand, are based on the temporal dependency structures introduced in [Zhang and Xue, 2018b], which includes both time expressions and events. [Kolomiyets et al., 2012] developed traditional statistical parsers specifically for narrative domain with extensive feature engineerings. We proposed more broad-coverage neural parsers with minimal domain-specific feature engineerings. These parsers are experimented and validated on two domains: narrative discourses in fairy tales and reporting discourses in the news domain. Experimental results in [Kolomiyets et al., 2012] show that their statistical parsers are better at temporal relation label prediction than a simple baseline, but didn't provide a significant improvement on temporal structure prediction, indicating that temporal structure is a more difficult task. Our proposed neural parsers significantly improved the performance on both temporal relation classification and temporal structure parsing.

2.2.4 A Comparative Analysis of Existing Temporal Models

This section gives an overview of different types temporal models described in the previous sections and discusses a comparative analysis of major similarities and differences between these temporal models. Particularly, our comparative analysis will focus on the following temporal models with accompanying annotated corpora (if any): TimeML (Timebank, Timebank-Dense), Multi-axis Temporal Model (MATRES), Narrative Containers (THYME), Temporal Discourse Model, Temporal Dependency Tree structure (Chinese and English TDT corpora).

In order to give a qualitative and intuitive comparison, we would like to examine the different temporal models annotated on the same document. Timebank-Dense is the most commonly annotated corpus among these temporal models. Therefore we picked the shortest document from Timebank-Dense (ea980120.1830.045) and collected its temporal model annotations on all of these temporal models except for TDM and NC which we annotated ourselves according to their guidelines. The document is presented as follows in (9), with TimeML Timex3s marked in orange, TimeML events in blue, and TimeML signals in green. Figure 2.1 $\sim$ Figure 2.6 illustrate different temporal models for this document.

(9): DCT: 1998-01-20$_{t13}$

The Pentagon said$_{e1}$ today$_{t14}$ it will re-examine$_{e2}$ the question$_{e17}$ are the remains inside the Tomb of the Unknown from the Vietnam War, in fact, known$_{e5}$?

CBS News first reported$_{e6}$ last night$_{t15}$ that the tomb may contain$_{e7}$ the remains of Air Force pilot Michael Blassie.

There was a suspicion$_{e18}$ the body was$_{e20}$ Blassie because his uniform and ID card were found$_{e9}$ near the body in Vietnam. But subsequently$_{s20}$, they were lost$_{e10}$. Blassie's mother

now$_{t16}$ wants$_{e11}$ the remains in the tomb tested$_{e12}$ for DNA.

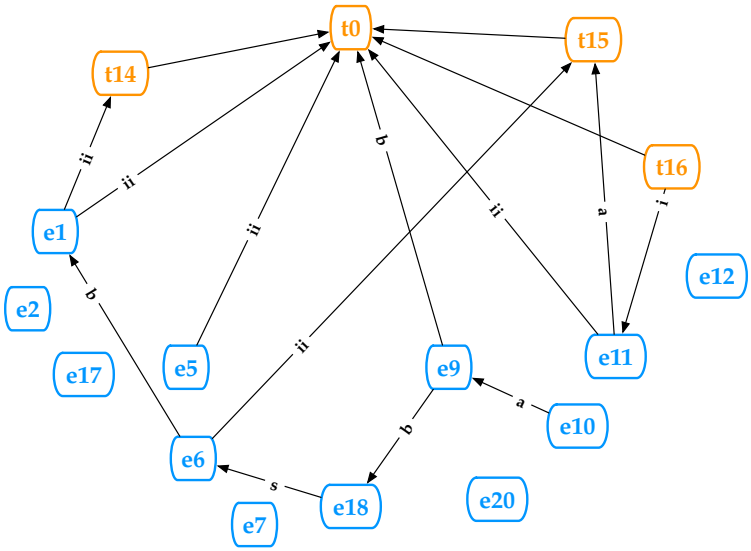


Figure 2.1: Timebank annotation for document (9).

Figure 2.1 and Figure 2.2 represent the annotations in Timebank and Timebank-Dense respectively. Red edges and edge labels in Figure 2.2 represent differences between the two annotations. In Timebank, annotators were allowed to pick temporal related event/time pairs with their own judgement and annotate relations for those pairs only. In Timebank-Dense, every pair of event/time in two adjacent sentence are required to be annotated with a temporal relation. As shown in the figures, while annotators consider only a few pairs of event/time hold valid temporal relations in Timebank, the Timebank-Dense guidelines greatly increased the number annotated relations, rendering a more laborious annotation yet more complete coverage. Both Timebank and Timebank-Dense are representative pair-wise models. As seen in the figures, pair-wise models are computationally represented as graphs. Timebank models each text as a partially-connected graph with some disconnected nodes/subgraphs, while Timebank-Dense models each text as a partially-connected graph with no disconnected subgraphs.

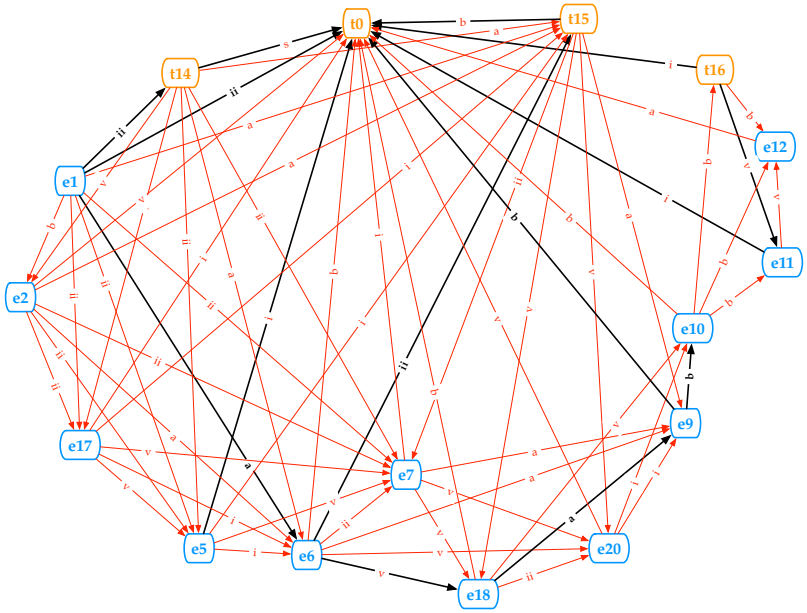


Figure 2.2: Timebank-Dense annotation for document (9).

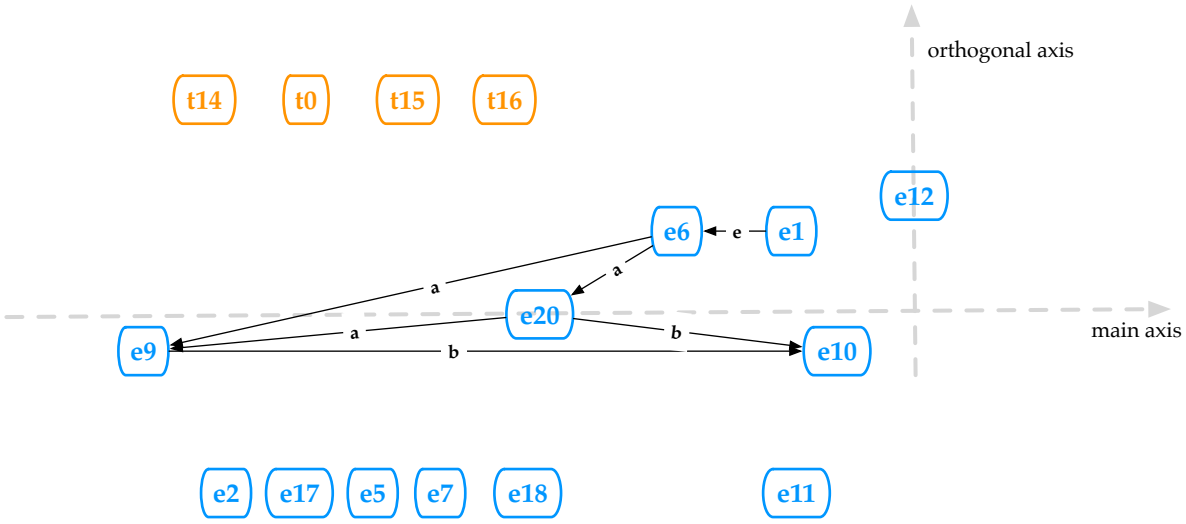
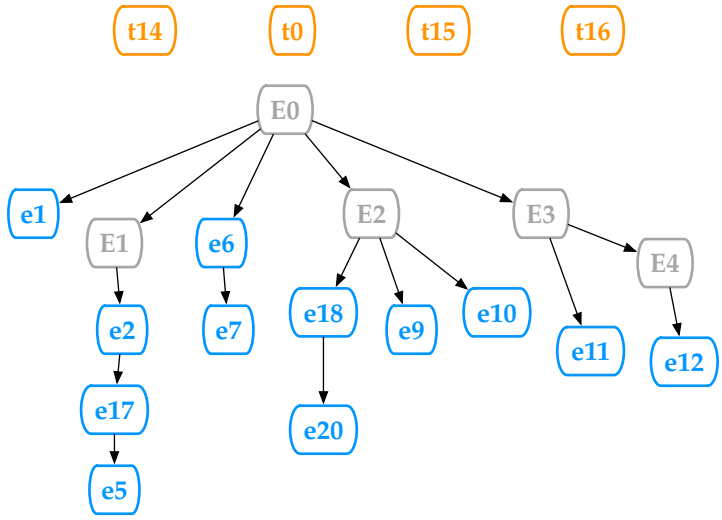


Figure 2.3: MATRES annotation for document (9).

Figure 2.3 illustrates an example for Multi-axis Temporal Model from the MATRES corpus. As shown in the figure, the Multi-axis Temporal Model is also a partially-connected graph for a piece of text, with disconnected nodes/subgraphs. However, on top of the graph structure, Multi-axis

Model also structures certain disconnected subgraphs on different axes. Note that this model doesn't cover temporal relations with regards to time expressions.



{e1 < e2, e17 ⊆ e2, e5 ⊆ e17,
e6 < e1, e7 ⊆ e6,
e9 < e18, e20 ⊆ e18, e18 < e10, e10 < e6,
e10 < e11, e11 < e12}

Figure 2.4: TDM annotation for document (9) (our own annotation).

Figure 2.4 shows the temporal structure of the same document in a Temporal Discourse Model. Unlike previous graph models, a TDM models temporal information in the text as a tree structure, with events in the text and some abstract events as nodes, and "temporal inclusion" relation as edges. Temporal ordering relations are modeled as a separate set of constraints. Like the Multi-axis Model, TDM doesn't model temporal relations with regards to time expressions.

Figure 2.5 illustrates the temporal structure of the same document in a Narrative Container model. With a few DCT related abstract nodes, this model is also a partially-connected graph with some disconnected nodes/subgraphs. It represents time expressions and events on nodes, temporal "contain" relations on edges between narrative containers and events, and temporal ordering relations on edges among events within the same narrative container. One event can also belong to multiple

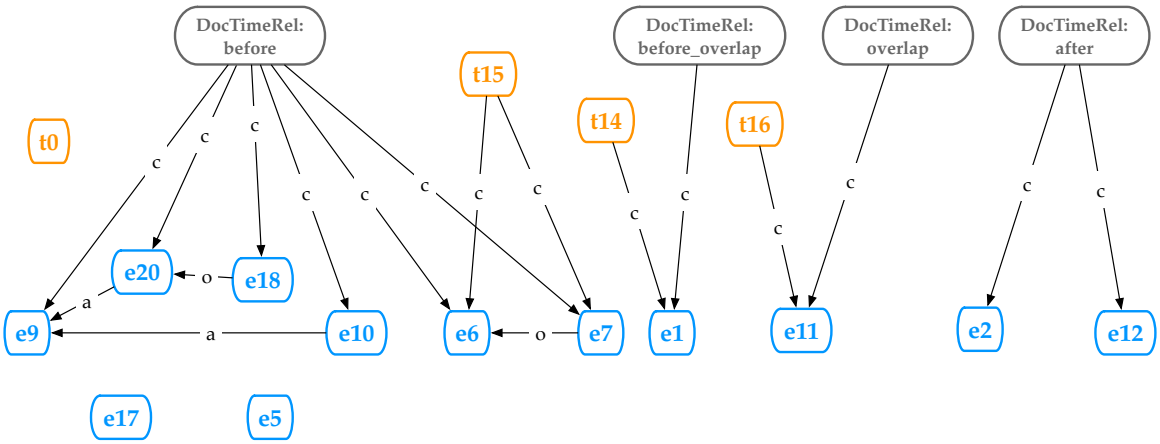


Figure 2.5: Narrative Container annotation for document (9) (our own annotation).

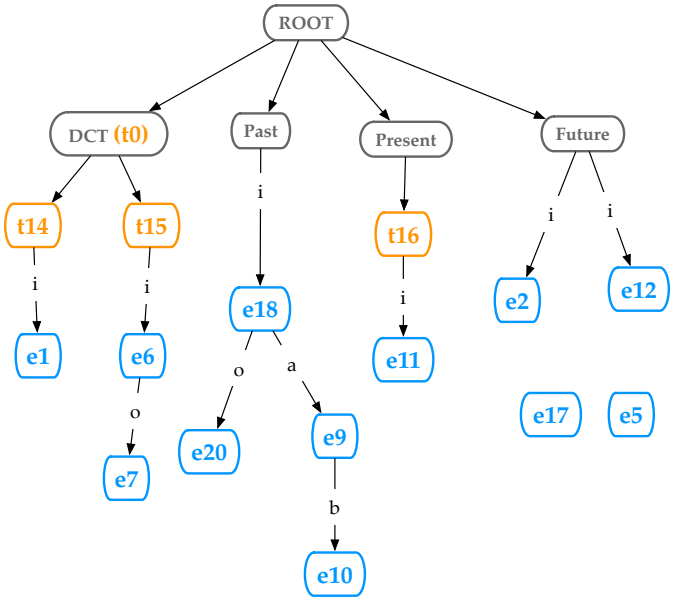


Figure 2.6: TDT annotation for document (9).

narrative containers.

Lastly, our TDT model of the same document is represented in Figure 2.6. It is an acyclic tree structure with a few abstract nodes on the top layers. It represents time expressions and events on nodes, and temporal anaphora and temporal relations on edges.

For a quantitative comparison, we compiled statistics and annotation agreements of these corpora in the following tables. Table 2.9 illustrates statistics on number of documents, timex, events, and temporal relations on these corpora. For MATRES, statistics on orthogonal axes (i.e. intention and opinion axes) are in round parentheses, while the other statistics are for the main axis only. THYME (total) is the entire THYME corpus, including both colon and brain cancer domains, and THYME (1st release) is the first released THYME data introduced in [Styler IV et al., 2014b], a subset of THYME corpus including only some of the colon cancer data.

Table 2.10 and Table 2.11 give the inter-annotator agreements for each corpus. Please note that these numbers are not necessarily directly comparable since they sometimes evaluate different specific agreements. The F1 scores here are computed with picking one expert as the gold standard, and for temporal relations evaluations, this F1 should be the same with P and R. The kappa scores here are either Cohen's kappa or Fleiss' kappa depending on the number of annotators involved. Crowdsourcing worker accuracies are computed against expert annotations, and WAWA is Worker Agreement With Aggregate measure among crowdsource workers. For MATRES corpus, all agreements reported here are for the main-axis only, and expert agreements are computed on a subset of Timebank-Dense (100 events and 400 relations), while crowd worker agreements are computed on Timebank-Dense.

Here we look at two different evaluations. The first evaluation is annotators' agreement on determining the pair of event/time that needs a temporal relation annotation (Table 2.10). This measure is slightly different for different models. Specifically, for Timebank and TB-Dense, these numbers evaluate annotators' agreements on judging if a temporal relation should be annotated between a given pair of timex/events. For MATRES, they evaluate annotators' agreements on whether an

Temporal Model	Corpus	Domain	# Docs	# Timex	# Events	# Temporal Relations
TimeML	Timebank	news	183	1,414	7,935	6,418
	Timebank-Dense	news	36	289	1,729	12,715
Multi-axis	MATRES	news	276	-	5,453*	13,577*
	MATRES (TB-Dense)		36	-	544* (69)	1,673* (128)
NC	annotations on Timebank	news	183	1,414	7,935	7,935 ***
	THYME (total)	clinical notes	1,186	14,440	127,736	30,545 ****
	THYME (1st release)		107	1,426	15,769	7,935
TDM	Remedia, BRC, CBC **	grade school reading materials	1,200+	-	-	-
TDT	Chinese TDT	news	115	1,167	4,807	5,974
		fairy tales	120	131	10,976	11,107
	English TDT	news	183	1,414	2,691	4,105

Table 2.9: Corpora Statistics. (* Stats on main axis only; numbers in parentheses are for orthogonal axes. ** Brandeis Reading Comprehension corpus, and Canadian Broadcasting Corporation corpus. *** This statistic is not reported in their paper; however, it should be the same number as the number events according to their annotation approach. **** This number reports only the "contains" relation.)

Temporal Model	Corpus	Experts		Crowd Workers	
		F1	Kappa	ACC	WAWA
TimeML	Timebank	.55	-	-	-
	Timebank-Dense	-	-	-	-
Multi-axis	MATRES	-	.85	.86	.79
NC	annotations on Timebank	-	-	-	-
	THYME (total)	-	-	-	-
	THYME (1st release)	.50	-	-	-
TDM	Remedia, BRC, CBC	-	-	-	-
TDT	Chinese TDT (news)	.86	-	-	-
	Chinese TDT (fairy tales)	.83	-	-	-
	English TDT	-	-	.82	.81

Table 2.10: IAAs on Pair Extraction Annotations.

event is anchorable on a given axis. For NC, they evaluate annotators' agreements on selecting the narrative container or narrative time for a given event. For TDT, they evaluate annotators' agreements on selecting the reference time / parent for a given event. The second evaluation is temporal relation annotation when pairs of time/event are given (Table 2.11). This measure is slightly dif-

Temporal Model	Corpus	Experts		Crowd Workers	
		F1	Kappa	ACC	WAWA
TimeML	Timebank	.77	.71	-	-
	Timebank-Dense	.65∼.72	.56∼.64	-	-
Multi-axis	MATRES	.90	.84	.88	.81
NC	annotations on Timebank	-	(.74)	-	-
	THYME (total)	({.66}) ({.80*}) [.52] [.71*]	-	-	-
	THYME (1st release)	(.45) ({.56}) [.72]	-	-	-
TDM	Remedia, BRC, CBC	-	-	-	-
TDT	Chinese TDT (news)	(.79)	-	-	-
	Chinese TDT (fairy tales)	(.72)	-	-	-
	English TDT	-	-	.83 (.53)	.85 (.52)

Table 2.11: IAAs on Temporal Relation Annotations. These numbers evaluate annotators' agreements on labeling the temporal relation between a given pair of timex/events. (Numbers in parentheses are NOT relation only evaluations; they evaluate both pair extraction & relation labeling together. Numbers in square brackets report only temporal relations with respect to DCT. Numbers in curly brackets report only on the "contains" temporal relation. * These numbers report agreements between annotator majority and adjudicator.)

ferent for different models as well. For some corpora, agreement on temporal relation only is not available and instead the evaluation for both pair extraction and relation labeling together are given. These numbers are presented in round parentheses. THYME annotation was mainly focused on event-document temporal relations (marked in square brackets in the table) and "contains" relation between events and times (marked in curly brackets).

Chapter 3

Structured Interpretation of Temporal Relations

3.1 Introduction

Understanding temporal relations between events and temporal expressions in a natural language text is a fundamental part of understanding the meaning of text. Automatic detection of temporal relations also enhances downstream natural language applications such as story timeline construction, question answering, text summarization, information extraction, and others. Due to its potential, temporal relation detection has received a significant amount of interest in the NLP community in recent years.

Most of the research attention has been devoted to defining the "semantic" aspect of this problem – the identification of a set of semantic relations between pairs of events, between an event and a time expression, or between pairs of time expressions. Representative work in this vein includes

TimeML [Pustejovsky et al., 2003a], a rich temporal relation markup language that is based on and extends Allen's Interval Algebra [Allen, 1984]. TimeML has been further enriched and extended for annotation in other domains [O'Gorman et al., 2016, Styler IV et al., 2014a, Mostafazadeh et al., 2016]. Corpora annotated with these schemes [Pustejovsky et al., 2003b, O'Gorman et al., 2016] are shown to have stable inter-annotator agreements, validating the temporal relations proposed in the TimeML. Through a series of TempEval shared tasks [Verhagen et al., 2007a, Verhagen et al., 2010a, UzZaman et al., 2012, Bethard et al., 2015a, Bethard et al., 2016a, Bethard et al., 2017], there has also been significant amount of research on building automatic systems aimed at predicting temporal relations.

Less attention, however, has been given to the "structural" aspect of temporal relation modeling – answering the question of which other events or time expressions a given time expression or event depends on for the interpretation of its temporal location. Having an answer to this question is important to both linguistic annotation and computational modeling. From the point of view of linguistic annotation, without an answer to this question, an annotator is faced with the choice of: (i) labeling the relation between this event/time expression with all other events and time expressions, or (ii) choosing another event/time expression with which the event/time expression in question has the most salient temporal relation. (i) is impractical for any textual document that is longer than a small number of sentences. Without a solid linguistic foundation, adopting (ii) could lead to inconsistent and incomplete annotation as annotators may not agree on which temporal relations are the most salient.

From a computational perspective, without knowing which time expressions and events are related to each other, an automatic system has to make a similar choice to predict the temporal relations between either all pairs of events and time expressions, or only a subset of the temporal relations. If it chooses to do the former, there will be $\binom{n}{2}$ pairs for n events and time expressions. Not only

is this computationally expensive, there could be conflicting predictions due to the transitivity of temporal relations (e.g. “A before B" and “B before C" imply “A before C", which a pair-wise approach may make conflicting predictions) and additional steps are necessary to resolve such conflicts [Chambers and Jurafsky, 2008b, Yoshikawa et al., 2009, Do et al., 2012].

We propose a novel annotation approach to address this dilemma. Specifically we propose to build a dependency tree structure for the entire document where the nodes of the tree are events and time expressions, as well as a few pre-defined “meta” nodes that are not anchored to a span of text in the document. The building blocks of this dependency structure are pairs of events and time expressions in which the *child* event/time expression depends on its *parent* event/time expression for its temporal interpretation. The dependency relation is based on the well-established notion of temporal anaphora where an event or time expression can only be interpreted with respect to its reference time [Reichenbach, 1947, Partee, 1973, Partes, 1984, Hinrichs, 1986, Webber, 1988, Bohnemeyer, 2009]. In each dependency relation in our dependency structure, the parent is the *antecedent* and the child is the *anaphor* that depends on its antecedent for its temporal interpretation. Consider the following examples:

(10): He arrived on Thursday. He got here at 8:00am.

(11): He arrived at school, walked to his classroom, and then the class began.

In (10), the antecedent is “Thursday” while “8:00am” is the anaphor. We won’t know when exactly he arrived unless we know the 8:00am is on Thursday. In this sense, “8:00am” depends on “Thursday” for its temporal interpretation. We define the *antecedent of an event as a time expression or event with reference to which the temporal location of the anaphor event can be most precisely determined.* In (11), the antecedent for the event “the class began” is “walked to his classroom” in the sense that the most specific temporal location for the event “the class began” is after he walked

to the classroom. Although "the class began" is also after "he arrived at school", the temporal location we can determine based on that is not as precise.

In order for the events and time expressions to form a dependency tree, one key assumption we make is *there is exactly one antecedent event/time expression for each anaphor*. This ensures that there is exactly one head for each dependent, a key formal condition for a dependency tree.

Once this dependency structure is acquired, manually or automatically, additional temporal relations may be inferred based on the transitive property of temporal relations, but we argue that this dependency structure is an intuitive starting point that makes annotation as well as the computational modeling more constrained and tractable.

We annotate a corpus of 235 documents with temporal dependency structures, with 48 documents double-annotated to evaluate inter-annotator agreement. The annotated data are chosen from two different genres, news data from the Xinhua newswire portion of the Chinese TreeBank [Xue et al., 2005] and Wikipedia news data used for CoNLL Shared Task on Shallow Discourse Parsing in 2016 [Xue et al., 2016], and narrative story data from Grimm fairy tales. The two genres are chosen because the temporal structure of texts from those two genres unfolds in very different ways: news reports are primarily in report discourse mode in the sense of [Smith, 2003] while Grimm fairy tales are primarily in narrative mode and time advances in those two genres in very different ways, as we will discuss in more details in Section 3.3.2.2. We report a stable and high inter-annotator agreement for both genres, which validates the intuitiveness of our approach. This corpus is publicly available.[1]

The main contributions of this chapter are:

- We propose a novel and comprehensive temporal dependency structure to capture temporal

[1] https://github.com/yuchenz/structured_temporal_relations_corpus

relations in text.

- We analyze different types of time expressions in depth and propose a novel definition, as far as we know, for the reference time of a time expression (§3.2.2.1).

- We produce an annotated corpus with this temporal structure that covers two very different genres, news and narratives and achieved high inter-annotator agreements for each genre. An analysis of the annotated data show that temporal structures are very genre-dependent, a conclusion that has implications for how the temporal structure of a text can be parsed.

3.2 Temporal Structure Annotation Scheme

In our annotation scheme, a temporal dependency tree structure is defined as a 4-tuple (T, E, N, L), where T is a set of time expressions, E is a set of events, and N is a set of pre-defined "meta" nodes not anchored to a span of text in the document. T, E, N form the nodes in the dependency structure, and L is the set of edges in the tree. Figure 3.1 gives an example TDT. Detailed descriptions for each set are in the following subsections.

3.2.1 Nodes in the temporal dependency tree

The nodes in a temporal dependency tree includes time expressions, events, and a set of pre-defined nodes. We elaborate on each type of nodes below:

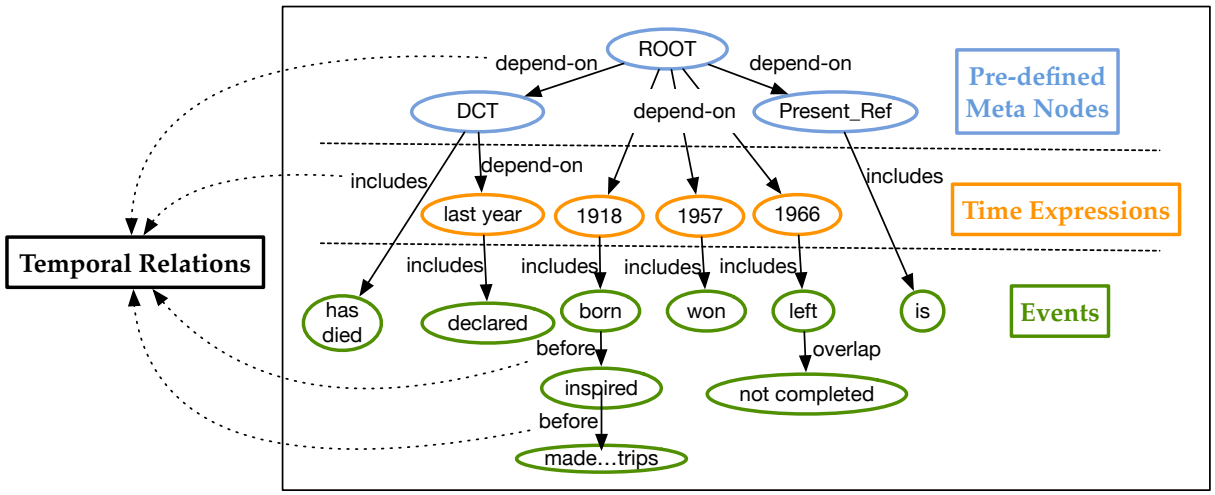


Figure 3.1: An example TDT.

3.2.1.1 Time Expressions

TimeML [Pustejovsky et al., 2003a] treats all temporal expressions as markable units and classifies them into three categories: fully specified temporal expressions ("June 11, 1989", "Summer, 2002"); underspecified temporal expressions ("Monday", "next month", "last year", "two days ago"); and durations ("three months", "two years"). The purpose of our dependency structure annotation is to find all time expressions that can serve as a reference time for other events or time expressions. We observe that while the first two TimeML categories of time expressions can serve as reference times, the last category, "durations", typically don't serve as reference times, unless they are modified by expressions like "ago" or "later". For example, the "10 minutes" in (12) can serve as a reference time because it can be located in a timeline as a duration from 8:00 to 8:10, while the "10 minutes" in (13) can't serve as a reference time.

(12): He arrived at 8:00am. 10 minutes later, the class began.

(13): It usually takes him 10 minutes to bike to school.

Therefore, in our annotation scheme, we make the distinction between time expressions that can be used as reference times and the ones that cannot. The former includes fully specified temporal expressions, underspecified temporal expressions, as well as time durations modified by "later" or "ago". The latter include unmodified durations. In our annotation, only the former are considered to be valid nodes in our time expression set T.

3.2.1.2 Events

We adopt a broad definition of events following [Pustejovsky et al., 2003a], where "an event is any situation (including a process or a state) that happens, occurs, or holds to be true or false during some time point (punctual) or time interval (durative)." Based on this definition, unless stated explicitly, events for us include both eventive and stative situations. Adopting the minimal span approach along the lines of [O'Gorman et al., 2016], only the headword of an event is labeled in actual annotation. Since different events tend to have different temporal behaviors in how they relate to other events or time expressions[Wuyun, 2016], we also assign a coarse event classification label to each event before linking them to other other events or time expressions to form a dependency structure. Adapting the inventory of situation entity types from [Smith, 2003] and from [Zhang and Xue, 2014], we define the following eight categories for events.

- An **Event** is a process that happens or occurs. It is the only eventive type in this classification set that advances the time in a text. An example event is "I *went* to school yesterday".
- A **State** is a situation that holds during some time interval. It is stative and describes some property or state of an object, a situation, or the world. For example, "she *was* very shy" describes a state.

The remaining event types are all statives that describe an eventive process.

- A **Habitual** event describes the state of a regularly repeating event, as in "I *go* to the gym three times a week".
- An **Ongoing** event describes an event in progress, as in "she was *walking* by right then".
- A **Completed** event describes the completed state of an event, as in "She's *finished* her talk already".
- A **Modalized** event describes the capability, possibility, or necessity of an event, as in "I have to *go*".
- A **Generic Habitual** event is a Habitual event for generic subjects, as in "The earth *goes* around the sun".
- A **Generic State** is a state that hold for a generic subject, as in "Naked mole rats don't *have* hairs".

  All valid events from a document, represented by their headwords, form the event set E.

3.2.1.3 Pre-defined Meta Nodes

In order to provide valid reference times for all events and time expressions, and to form a complete tree structure, we designate the following pre-defined nodes for the set N.

ROOT is the root node of the temporal dependency tree and every document has one ROOT node. It is the parent of (i) all other pre-defined nodes, and (ii) absolute concrete time expressions (Example 20, see §3.2.2.1 for more on time expression classification). The meta node **DCT** is

the Document Creation Time, a.k.a. Speech Time. Following [Pustejovsky et al., 2003a], we define meta nodes **PRESENT_REF, PAST_REF, FUTURE_REF** as the general reference times respectively for generic present, past, and future times. Lastly, **ATEMPORAL** is designated as the parent node for atemporal events, such as timeless generic statements (Example 21).

These generic reference times are necessary for time expressions and events that don't have a more specific reference time in the text as their parents. For example, it is common to start a narrative story with a few descriptive statements in past tense without a specific time (Example 14), or a general time expression referring to the past (Example 15). Both cases take "Past_Ref" as their parent.

(14): It was a snowy night. [Past_Ref]

(15): Once upon the time, ... [Past_Ref]

It is worth noting that "DCT" and "Present_Ref" are not interchangeable. "DCT" is usually a very specific time-stamp such as "2018-02-15:00:00:00", while "Present_Ref" is a general temporal location reference. We use "DCT" as the parent for relative concrete time expressions. For instance, in Example 19 below, the reference time for "last year" is "DCT" rather than "Present_Ref", because with the knowledge of DCT being, for example "2018-02-15:00:00:00", the temporal location of "last year" can be determined as "2017". Therefore, we designate that the interpretation of the temporal location of "last year" is dependent on "DCT", and "DCT" should be the parent of "last year". Yet for vague time expressions, such as Example 16, their antecedent should be "Present_Ref" . More details on time expression classification are described in §3.2.2.1.

(16): China annual economic output results have grown increasingly smooth in recent years. [Present_Ref]

(17): Economists who try to estimate actual growth tend to come up with lower numbers. [Present_Ref]

(18): China will remain a trade partner as important to Japan as the United States in the future. [Future_Ref]

(19): The economy expanded 6.9 percent last year. [DCT]

(20): A trend of gradual growth began in 2011. [ROOT]

(21): The earth goes around the sun. [Atemporal]

3.2.2 Edges in the temporal dependency tree

As we discussed above, each dependency relation consists of an antecedent and an anaphor, with the antecedent being the parent and the anaphor being the child. Based on the well-established notion of temporal anaphora [Reichenbach, 1947, Partee, 1973, Partes, 1984, Hinrichs, 1986, Webber, 1988, Bohnemeyer, 2009], we assume each event or time expression in the dependency tree has only one antecedent (i.e. one reference time), which is necessary to form the dependency tree. In this section, we will first discuss what can serve as a reference time for time expressions in our annotation scheme, then we will discuss what can be a reference time for events. All links between events/time expressions and their reference times form our link set L.

3.2.2.1 Reference Times for Time Expressions

In previous work such as the TimeBank [Pustejovsky et al., 2003a] the temporal relations between time expressions are annotated with temporal ordering relations such as “before”, “after”, or “overlap” just like events in a pair-wise without considering the dependencies between them.

For example, consider the three time expressions "2003", "March", and "next year" in (22), using a pair-wise annotation approach, three temporal relations will be extracted:

(2003, includes, March)

(2003, before, next year)

(March, before, next year)

(22): The economy expanded 6.6 percent in $\underline{2003}_{t1}$, reaching its peak 7.1 percent in $\underline{\text{March}}_{t2}$. The growth rate doubled in the $\underline{\text{next year}}_{t3}$.

We argue that the sole purpose for annotating temporal relations between time expressions is to properly "interpret" time expressions that "depend" on another time expression for their interpretation. In the context of time expressions, "interpretation" means normalizing time expressions in a format that allows the ordering between the time expressions to be automatically computed. Time expression normalization is necessary in many applications. For example, in a question answering system, our model needs to be able to answer "2004" when it is asked "Which year did China's export rate double?", instead of answering "next year" which is uninterpretable taken out of the original context. In order for the time expressions to be properly interpreted, it is important to annotate the dependency between "March" and its reference time "2004" because the former depends on the latter for its interpretation. Similarly, it is also important to establish the dependency between "next year" and its reference time "2004" as we won't know which year is "next year" until we know it is with reference to "2004". With these dependencies identified and the time expressions normalized, the temporal relations between all pairs of time expressions in a text can be automatically computed, and explicit annotation of the temporal relation between all pairs of time expressions will not be necessary. For example, with "March" normalized to "2003-03" and "next year" normalized to "2004", the relation between 2003-03 and 2004 can be automati-

cally computed. We argue that this notion of reference time for time expressions is intuitive and easy to define. Annotating temporal dependency relations between each pair of time expression and its parent (i.e. finding the reference time for each time expression) is also more efficient than annotating the temporal ordering between all pairs of time expressions.

Based on these considerations, we propose a novel definition of the reference time for time expressions:

Definition 3.2.1 *Time expression A is the reference time for time expression B, if B depends on A for its temporal location interpretation.*

In other words, a time expression can depend solely on its reference time to be interpreted and normalized. We use a generic **Depend-on** label for these relations. Take the following (23) as an example, annotators only need to determine that the temporal interpretation of '8am" depends on "Thursday". With "Thursday" normalized to, for example, "2003-04-05", we can then compute a normalized time "2003-04-05:08:00:00" for "8am", and easily compute the temporal ordering between them: ("2003-04-05" includes "2003-04-05:08:00:00").

(23): He arrived on Thursday. He got here at 8:00am.

We now consider the question of what types of nodes can serve as the reference time or antecedent for a time expression. First, since a time expression relies on its reference time for its temporal interpretation, naturally an event cannot serve as its reference time. Second, since some time expressions (e.g., "2003") can be interpreted (and normalized) on its own without any additional information, while others can not, further categorization of time expressions is needed to precisely specify which time expressions need a reference time for their interpretation and which do not, and what time expressions can serve as reference times and which do not.

Taxonomy	Time Expressions			
	Locatable Time Expressions			Unlocatable Time Expressions
	Concrete		Vague	
	Absolute	Relative		
Examples	May 2015	today, last Monday ... two days later	nowadays	every month
Possible Reference Times	ROOT	DCT, another Concrete	Present_Ref, Past_Ref, Future_Ref	-

Table 3.1: Taxonomy of time expressions in our annotation scheme, with examples and possible reference times.

First, we make the distinction between Concrete and Vague time expressions. A **Concrete Time Expression** is a time expression that can be located onto a timeline as an exact time point or interval, e.g. "June 11, 1989", "today". Their starting and ending temporal boundaries on the timeline can be determined. A **Vague Time Expression** (e.g., "nowadays", "recent years", "once upon the time") expresses the concept of (or a period in) general past, general present, or general future, without specific temporal location boundaries. The reference time for Vague time expressions are the pre-defined nodes PRESENT_REF, PAST_REF, and FUTURE_REF.

Concrete time expressions are further classified into **Absolute Time Expressions** and **Relative Time Expressions**, corresponding to fully-specified ("June 11, 1989", "Summer, 2002") and underspecified temporal expressions ("Monday", "Next month", "Last year", "Two days ago") in [Pustejovsky et al., 2003a] respectively. Relative concrete time expressions take either DCT or another concrete time expression as their reference time. Absolute concrete time expressions can be normalized independently and don't need a reference time. Therefore, we stipulate that their parent in the dependent tree is the pre-defined node ROOT. For example, "1995", "20th century" are absolute concrete time expressions, while "today", "last year", "the future three years", "January 20th", "next Wednesday" are relative concrete time expressions, and "recent years", "in the

past a few years", "nowadays", "once upon the time" are vague time expressions.

An example of a concrete relative time expression having a concrete absolute temporal expression as its reference time is given in (22) . Consider the time expression "March". In order to be able to interpret it and normalize it into a valid temporal location on a timeline, we need to establish "2003" is its reference time. Then it is possible to normalize it into a formal representation as "2003-03".

Lastly, in order to form a complete tree structure, all pre-defined nodes (except for ROOT) take ROOT as their parent. A complete taxonomy of time expressions in our annotation scheme with examples and their possible reference times is illustrated in Table 3.1. Note that in our framework, we simply exclude unlocatable time expressions, instead of linking them to the pre-defined meta node "Atemporal", which is designated as the parent for atemporal events.

3.2.2.2 Reference Times for Events

The reference time for an event is a time expression or pre-defined node or another event with respect to which the most specific temporal location of the event in question can be determined. Unlike time expressions, for which the possible reference times can only be other time expressions or pre-defined nodes, the possible reference times for events are not as restrictive and can be any of the three categories. The dependency relation that we use to characterize the relationship between the reference time / antecedent and an event is a temporal relation between them.

Definition 3.2.2 *Time expression/pre-defined node/event A is the reference time for event B, if A is the most specific temporal location which B depends on for its own temporal location interpretation.*

There has been significant amount of work attempting to characterize the temporal relationship between events, and between time expressions and events. One of the first attempts to model temporal relations is Allen's Interval Algebra theory [Allen, 1984]. This theory introduced a set of distinct and exhaustive temporal relations that can hold between two time intervals, which are further adapted and extended in [Pustejovsky et al., 2003a], THYME [Styler IV et al., 2014a], etc. [Mostafazadeh et al., 2016] gives a detailed comparison of these temporal relations sets. Mindful of the need to produce consistent annotation, and in line with the practice of some prior work such as the TempEval evaluations [Verhagen et al., 2007b, Verhagen et al., 2009, Verhagen et al., 2010b], we adopt a simplified set of 4 temporal relations to characterize the relationship between an event and its reference time. The set of temporal relations we use with their mappings to their corresponding TimeML temporal relations are shown shown in Table 3.2.

Our Scheme	**TimeML**
Before	Before, IBefore
After	-
Overlap	Ends, Begins, Identity, Simultaneous
Includes	During

Table 3.2: Our temporal relation set for events with mappings to TimeML's set.

Although an event can in principle take a time expression, another event, or a pre-defined node as its antecedent, different types of events have different tendencies as to the types of antecedents they take. An eventive event usually takes either a time expression, another eventive event, or DCT as its reference time. They advance the time in the narrative of a text, so it usually has a (time expression, Includes, event) relation with its antecedent, or a (event, Before, event) relation. For example, in (10) the time expression "Thursday" has "Includes" relation with the event "arrived", and the time expression "8:00am" has an "Includes" relation with the event "got here". And in

Taxonomy	Events		
	Eventive	Stative	Generic
Examples	He arrived.	He was holding a book.	Planets go around stars.
Possible Reference Times	timex, DCT, eventive event, ~~stative event~~ Past/Future_Ref	timex, DCT, eventive event, stative event, Past/Present/Future_Ref	Atemporal

Table 3.3: Taxonomy of events in our annotation scheme, with examples and possible reference times.

(11) the event "arrived" has a "Before" relation with the event "walked".

A stative event can take a time expression, another event, or a pre-defined node (except for ROOT) as its reference time. It generally describes a state that holds during the time indicated by its antecedent time expression, event, or generic time. It usually has an "Overlap" relation with their reference times. For example, in (13) the event "takes" is a stative Habitual event, which describes a state of the present situation for "him", so its reference time is the pre-defined node "Present_Ref", and has an "Overlaps" relation with "Present_Ref".

An eventive event rarely takes a stative event as its reference time. As discussed above, we pick the most specific temporal location as the reference time for an event. Since more specific temporal locations are usually available (such as another eventive event), a stative event rarely serves as the reference time for an eventive event. Table 3.3 shows some of the most common event-parent scenarios.

Please see Appendix A: our annotation guidelines on time expression and event recognition, classification, and reference time resolution, for a complete account of specifications, examples, and special rules for special scenarios.

3.2.3 Full Temporal Structure Examples

We present a full example temporal dependency structure for a short news report paragraph (24), as illustrated in Figure 3.2, and another one for a narrative passage (25), as illustrated in Figure 3.3. Subscript e denotes eventive events, t denotes time expressions, and s denotes stative events. Please note that in our framework, nominalized events (e.g. "competition" in (24)) and events in relative clauses (e.g. "designed" in (24)) are currently not included in the scope of this work. Given six pre-defined meta nodes (blue in the figures), locatable time expressions (orange), and main events (green) in a text, an annotator will determine the most specific reference time for each time and event, and assign a temporal relation between the time/event and its parent. For example, when considering the "not completed" event in (24), we can see that it's a state that holds during the "left" event and also within the "1966" time period. However, "left" is determined to the final reference time for "not completed" because it's more specific than "1966", and locates the "not completed" state to a more accurate temporal location on the timeline. After all, according to only this piece of text, we can not tell whether or not "his plans for the interior of the building" was "completed" after he "left" but before the end of "1966".

(24): Jorn Utzon, the Danish architect who designed the Sydney Opera House, has died$_{e1}$ in Copenhagen. Born$_{e2}$ in 1918$_{t1}$, Mr Utzon was inspired$_{e3}$ by Scandinavian functionalism in architecture, but made a number of inspirational trips$_{e4}$, including to Mexico and Morocco. In 1957$_{t2}$, Mr Utzon's now-iconic shell-like design for the Opera House unexpectedly won$_{e5}$ a state government competition for the site on Bennelong Point on Sydney Harbour. However, he left$_{e6}$ the project in 1966$_{t3}$. His plans for the interior of the building were not completed$_{s1}$. The Sydney Opera House is$_{s2}$ one of the world's most classic modern buildings and a landmark Australian

structure. It was declared$_{e7}$ a UNESCO World Heritage site last year$_{t4}$. [2]

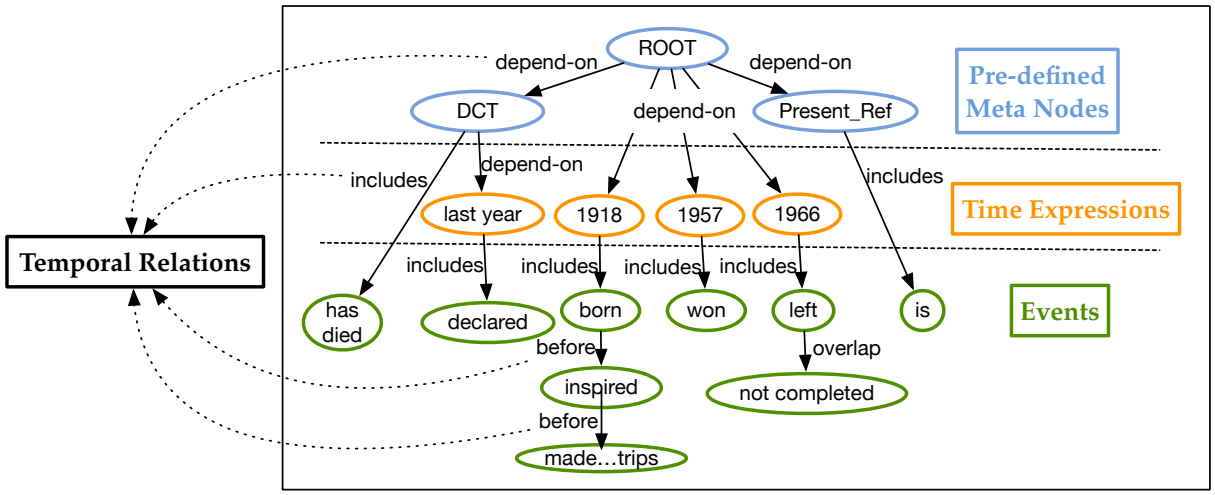


Figure 3.2: An example full temporal dependency structure for news paragraph (24).

(25): There was$_{s1}$ once$_{t1}$ a man who had seven sons, and still he had$_{s2}$ no daughter, however much he wished$_{s3}$ for one. At length his wife again gave$_{e1}$ him hope of a child, and when it came$_{e2}$ into the world it was$_{s4}$ a girl. The joy was$_{s5}$ great, but the child was$_{s6}$ sickly and small, and had to be privately baptized$_{s7}$ on account of its weakness. The father sent$_{e4}$ one of the boys in haste to the spring to fetch water for the baptism. The other six went$_{e5}$ with him, and as each of them wanted to be first to fill it, the jug fell$_{e6}$ into the well. There they stood$_{s8}$ and did not know$_{s9}$ what to do, and none of them dared to go$_{s10}$ home. As they still did not return, the father grew$_{e7}$ impatient, and said$_{e8}$, they have certainly forgotten$_{s11}$ it while playing some game, the wicked boys. He became$_{e9}$ afraid that the girl would have to die without being baptized.[3]

The two examples provide a sharp contrast between the typical temporal dependency structures for newswire documents and narrative stories, with the former generally having a flat and shallow structure and the latter having a narrow and deep structure.

[2]From a news report on *The Telegraph*
[3]From Grimm's fairy tale *The Seven Ravens*

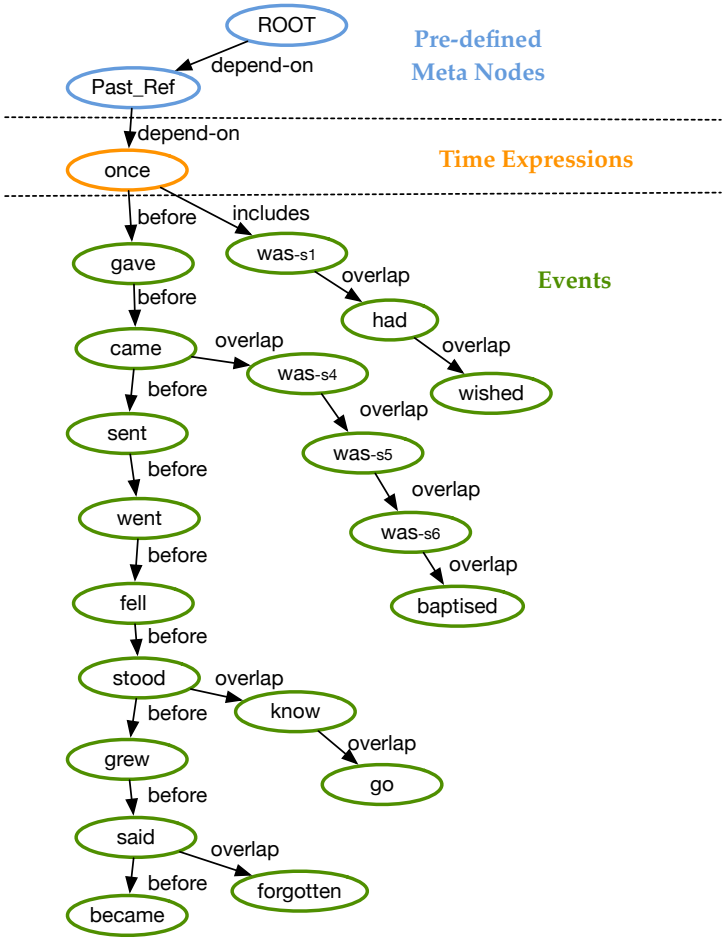


Figure 3.3: An example full temporal dependency structure for narrative paragraph (25).

3.3 Corpus Description and Analysis

3.3.1 Annotation Process

We use a two-pass annotation process for this project. In the first pass, annotators do temporal expression recognition and classification, and then reference time resolution for all time expressions. The purpose of this pass is to mark out all possible reference times realized by time expressions and recognize their internal temporal relations, in order to provide a backbone structure for the

final dependency tree. In the second pass, event recognition and classification, and then reference time resolutions for all events are annotated, completing the final temporal dependency structure of the entire document.

3.3.2 Annotation Analysis

3.3.2.1 Corpus

A corpus of 115 news articles, sampled from Chinese TempEval2 data [Verhagen et al., 2010a] and Wikinews data, [4] and 120 story articles, sampled from Chinese Grimm fairy tales, [5] are compiled and annotated. 20% of the documents are double annotated by native Chinese speakers. Table 3.4 presents the detailed statistics. High and stable inter-annotator agreements are reported in Table 3.5.

		# Docs	# Sentences	# Tokens	# Timex	# Events
News	Single	91	2,271	45,132	901	3,758
	Double	24	570	11,132	265	1,047
	Total	115	2,841	56,264	1,166	4,805
Narratives	Single	96	2,903	77,299	92	8,362
	Double	24	797	17,456	40	1,952
	Total	120	3,700	94,755	132	10,314

Table 3.4: Corpus annotation statistics. (*Timex* stands for time expressions.)

On event annotation, our work is comparable to the annotation work in [Kolomiyets et al., 2012]. They report inter-annotator agreements of 0.86, 0.82, and 0.70 on event recognition, unlabeled relations, and labeled relations respectively on a narrative data. We argue that the comparable

[4] zh.wikinews.org

[5] https://www.grimmstories.com/zh/grimm_tonghua/index

or better agreements on narratives as shown in Table 3.5 show that incorporating the notion of linguistic temporal anaphora helps annotators make more consistent decisions. High (above 90%) agreements on time expression recognition and parsing indicate that our new definition of the reference time for time expressions is clear and easy for annotators to operate on. While event annotations receive lower agreements than time expressions on both genres, they are in general easier on news than on narratives, especially for event reference time resolution and edge labeling.

		News	Narratives
Timex	Recognition	.97	1.
	Classification	.95	.94
	Parsing	.93	.94
Event	Recognition	.94*	.93*
	Classification	.77	.75
	Relations (unlabeled)	.86	.83
	Relations (labeled)	.79	.72

Table 3.5: Inter-Annotator Agreement F scores on 20% of the annotations. (* This annotation focuses on main events only, excluding nominalized events and events in relative clauses.)

3.3.2.2 Analysis Across Different Genres

During our annotation, we discovered that narrative texts are very different from news with respect to their temporal structures. First, news texts are usually organized with abundant temporal locations, while narrative texts tend to start with a few temporal locations setting the scene and proceed with only events. As shown in Table 3.4, around 20% (1166) nodes in the news data are time expressions and 80% (4805) are event nodes, while in the narrative data the ratio of time expressions to events are 1% / 99% (132/10314). Table 3.6 shows that news articles have a higher portion (51%) of relative concrete time expressions, while narrative stories tend to use more (67%) vague time expressions.

Second, descriptive statements are more common in news data than in narratives, while long chains of time advancing eventives are more common in narratives. We can see from Table 3.7 that in news data only 30% events are eventive, leaving the rest 70% stative descriptions, while in narrative data over half of the events (51%) are eventive. From Table 3.8 we can also see that the major temporal relation in news is "overlap" (54%), representing dominative stative statements in reporting discourse mode, while narrative texts are dominated by the "before" relation (53%), with eventive statements advancing the story line.

Timex type	News	Narratives
Absolute Concrete	313 (27%)	16 (14%)
Relative Concrete	598 (51%)	20 (17%)
Vague	256 (22%)	79 (67%)

Table 3.6: Distribution of time expression types.

Event type	News	Narratives
Event	1457 (30%)	5594 (51%)
State	1802 (37%)	3366 (31%)
Habitual	102 (2%)	459 (4%)
Modalized	321 (7%)	458 (4%)
Completed	1041 (22%)	900 (8%)
Ongoing Event	80 (2%)	175 (2%)
Generic State	1 (0%)	17 (0%)
Generic Habitual	2 (0%)	5 (0%)

Table 3.7: Distribution of event types.

Another difference is that statives serve different major roles in news and narrative texts. News tend to have deep branches of overlapping statives with a time expression, DCT, or a general present/past/future reference time as their parent (descriptive statements as discussed above). Narrative texts have much less such long stative branches, however, they tend to have numerous short

Edge label	News	Narratives
Includes	1096 (18%)	157 (1%)
Before(After)	507 (8%)	5885 (53%)
Overlap	3246 (54%)	4914 (44%)
Depend-on	1125 (19%)	151 (1%)

Table 3.8: Distribution of temporal relations.

		Pre-defined Node	Time Expression	Eventive Event	Stative Event
News	Time Expression	1078 (92%)	89 (8%)	0	0
	Eventive Event	103 (9%)	290 (26%)	716 (65%)	0
	Stative Event	149 (8%)	192 (11%)	432 (24%)	1029 (57%)
Narratives	Time Expression	95 (83%)	20 (17%)	0	0
	Eventive Event	20 (0%)	25 (1%)	4875 (99%)	0
	Stative Event	25 (1%)	74 (2%)	1655 (49%)	1612 (48%)

Table 3.9: Distribution of parent types for each child type. Rows represent child types, and columns represent parent types.

branches of statives with an eventive event as their parent. These statives serve as the event's accompanying situations. For example, in (25) "was$_{s4}$", "was$_{s5}$", "was$_{s6}$", and "baptised$_{s7}$" are accompanying statives to "came$_{e2}$", describing the baby and the family and the situation they were in at that time. For each type of node, we compiled the distribution of its possible types of parent, shown in Table 3.9. It's worth noting that more than twice as much statives in news have a stative parent (57%) than the ones having an eventive parent (24%), contributing to deep stative branches, while in narratives a much higher percentage of statives directly depend on an eventive (49%), contributing to a large number of short stative branches.

These different temporal properties of news and narratives further result in shallow dependency structures for news texts with larger number of branches on the root node, yet deep structures for narrative texts with fewer but long branches. These differences are illustrated intuitively on Figure 3.2 and Figure 3.3.

3.4 Conclusion

In this chapter, we introduced a new representation to model temporal information in a document – the Temporal Dependency Tree (TDT) structure representation. We argue that this structure is linguistically intuitive, and is amenable to computational modeling. High and stable inter-annotator agreements in our annotation experiments provide further evidence supporting our structured approach to temporal interpretation. In addition, a significant number of documents covering two genres have been annotated. This corpus is publicly available for research on temporal relation analysis, story timeline construction, as well as numerous other applications.

Chapter 4

Temporal Structure Parsing

4.1 Introduction

In this chapter, taking advantage of our data set annotated with temporal dependency structures in Chapter 3, we develop a neural temporal dependency structure parser using minimal hand-crafted linguistic features. One of the advantages of neural network based models is that they are readily adaptable to new domains without further domain-specific feature engineering. We demonstrate this advantage by evaluating our temporal dependency parser on data from two domains: news reports and narrative stories. Our results show that our model beats a strong logistic regression baseline. Direct comparison with existing models is impossible because the only similar dataset used in previous work [Kolomiyets et al., 2012] that we are aware of is not available to us, but we show that our models are competitive against similar systems reported in the literature.

The main contributions of this chapter are:

- We design and build the first end-to-end neural temporal dependency parser. The parser is based on a novel neural ranking model that takes a raw text as input, extracts events and time expressions, and arranges them in a temporal dependency structure.
- We evaluate the parser by performing experiments on data from two domains: news reports and narrative stories, and show that our parser is competitive against similar parsers. We also show the two domains have very different temporal structural patterns, an observation that we believe will be very valuable to future temporal parser development.

The rest of the chapter is organized as follows. In Section 4.2, we discuss related work and position our work in the literature context. We describe our end-to-end pipeline system in Section 4.3. The neural sequence labeling model for time expression and event recognition are described in Section 4.4, and details of the neural ranking model for temporal structure parsing are discussed in Section 4.5. In Section 4.6 we present and discuss our experimental results and we conclude this chapter in Section 4.7.

4.2 Related Work

4.2.1 Related Work on Temporal Relation Modeling

There is a significant amount of research on temporal relation extraction [Bethard et al., 2007, Bethard, 2013, Chambers and Jurafsky, 2008a, Chambers et al., 2014]. Most of the previous work models temporal relation extraction as pair-wise classification between individual pairs of events and/or time expressions. Some of the models also add a global reasoning step to local pair-wise classification, typically using Integer Linear Programming, to exploit the transitivity property

of temporal relations [Chambers and Jurafsky, 2008a]. Such a pair-wise classification approach is often dictated by the way the data is annotated. In most of the widely used temporal data sets [Pustejovsky et al., 2003b, Chambers et al., 2014, Styler IV et al., 2014a, O'Gorman et al., 2016, Mostafazadeh et al., 2016], temporal relations between individual pairs of events and/or time expressions are annotated independently of one another.

Our work is most closely related to that of [Kolomiyets et al., 2012], which also treats temporal relation modeling as temporal dependency structure parsing. However, their dependency structure, as described in [Bethard et al., 2012], is only over events, excluding time expressions which are an important source of temporal information, and it also excludes states, which makes the temporal dependency structure incomplete. We instead choose to develop our model based on the data set described in [Zhang and Xue, 2018c], which introduces a more comprehensive and linguistically grounded annotation scheme for temporal dependency structures. This structure includes both events and time expressions, and uses the linguistic notion of *temporal anaphora* to guide the annotation of the temporal dependency structure. Since in this temporal dependency structure each paraent-child pair is considered to be an instance of temporal anaphora, the parent is also called the *antecedent* and the child is also referred to as the *anaphor*. The corpus consists of data from two domains: news reports and narrative stories.

4.2.2 Related Work on Neural Dependency Parsing

Most prior work on neural dependency parsing is aimed at syntactic dependency parsing, i.e. parsing a sentence into a dependency tree that represents the syntactic relations among the words. Recent work on dependency parsing typically uses transition-based or graph-based architectures combined with contextual vector representations learned with recurrent neural networks (e.g. Bi-

LSTMs) [Kiperwasser and Goldberg, 2016].

In temporal dependency parsing, for each event or time expression, there is more than one other event or time expression that can serve as its reference time, while the most closely related one is selected as the gold standard reference time parent. This naturally falls into a ranking process where all possible reference times are ranked and the best is selected.

In this sense our neural ranking model for temporal dependency parsing is closely related to the neural ranking model for coreference resolution described in [Lee et al., 2017], both of which extract related spans of words (entity mentions for coreference resolution, and events or time expressions for temporal dependency parsing). However, our temporal dependency parsing model differs from Lee et al's coreference model in that, on one hand, the ranking model for coreference only needs to output the best candidate for each individual pairing and cluster all pairs that are coreferent to each other, while on the other hand, our ranking model for temporal dependency parsing needs to rank not only the candidate antecedents but also the relations between the antecedent and the anaphor. In addition, the model also adds connectivity and acyclic constraints in the decoding process to guarantee a tree-structured output.

4.3 A Pipeline System

We build a two-stage pipeline system to tackle this temporal structure parsing problem. The first stage performs event and time expression identification. In this stage, given a text as input, spans of words that indicate events or time expressions are identified and categorized. We model this stage as a sequence labeling process. A standard Bi-LSTM sequence model coupled with BIO labels is applied here. Word representations are the concatenation of word and POS tag embeddings.

The second stage performs the actual temporal structure parsing by identifying the antecedent for each time expression and event, and identifying the temporal relation between them. In this stage, given events and time expressions identified in the first stage as input, the model outputs a temporal dependency tree in which each child node is an event or time expression that is temporally related to another event or time expression or pre-defined meta node as its parent node. This stage is modeled as a ranking process: for each node, a finite set of neighboring nodes are first selected as its candidate parents. These candidates are then ranked with a neural network model and the highest ranking candidate is selected as its parent. We use a ranking model because it is simple, more intuitive, easier to train, and the learned model rarely makes mistakes that violate the structural constraint of a tree.

4.4 Stage1: Neural Sequence Labeling Model

We use a neural sequence labeling model for the first stage: time expression and event identification. For each sentence, a standard Bi-LSTM sequence model with BIO labeling is applied. The concatenation of word and POS tag embeddings are used as word representations. Model architecture is illustrated in Figure 4.1.

The **Forward Computation** is:

$$\boldsymbol{x}_k = [\boldsymbol{w}_k, \boldsymbol{pos}_k]$$

$$\boldsymbol{x}^*_k = BiLSTM(\boldsymbol{x}_k)$$

$$\boldsymbol{h}_k = tanh(\mathbf{W_1} \cdot \boldsymbol{x}_k + \mathbf{b_1})$$

$$\boldsymbol{o}_k = softmax(\boldsymbol{h}_i)$$

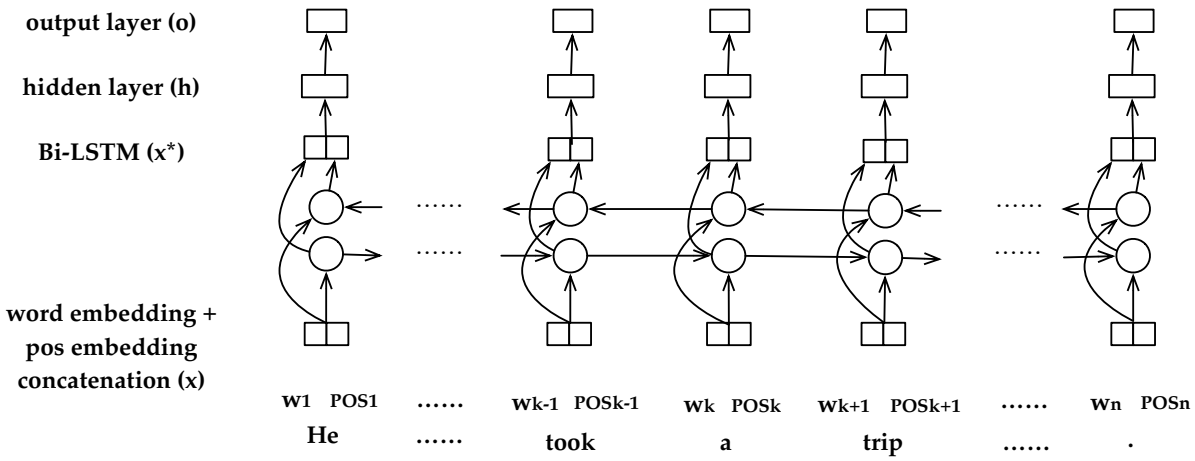


Figure 4.1: Neural Sequence Labeling Model Architecture.

Let D be the training data set of S sentences, N_s the number of words in sentence D_s, and y_i the gold BIO label for word i. The **Learning** process for our neural sequence labeling model tries to optimize the following cost function:

$$
\begin{aligned}
C &= -log \prod_{s}^{S} P(y_1, ..., y_{N_s} | D_s) \\
&= -log \prod_{s}^{S} \prod_{i}^{N_s} P(y_i | D_s) \\
&= \sum_{s}^{S} \sum_{i}^{N_s} -log P(y_i | D_s)
\end{aligned}
$$

For each training example i, cross-entropy loss is minimized:

$$
\begin{aligned}
L &= -log P(y_i | D_s) \\
&= -log \frac{exp[o_{y_i}]}{\sum_{y'_i} exp[o_{y'_i}]}
\end{aligned}
$$

where the concatenation of $o_{y'_i}$ forms $\boldsymbol{o}_i$ as described in the forward computation equations above.

4.5 Stage2: Neural Ranking Model

4.5.1 Model Description

We use a neural ranking model for the parsing stage. For each time expression or event node i in a text, a group of candidate parent nodes (time expressions, events, or pre-defined meta nodes) are selected. In practice, we select a window from the beginning of the text to two sentences after node i, and select all nodes in this window and all pre-defined meta nodes as the candidate parents if node i is an event. Since the parent of a time expression can only be a pre-defined meta node or another time expression as described in [Zhang and Xue, 2018c], we select all time expressions in the same window and all pre-defined meta nodes as the candidate parents if node i is a time expression. Let y'_i be a candidate parent of node i, a score is then computed for each pair of (i, y'_i).Through ranking, the candidate with the highest score is then selected as the final parent for node i.

Model architecture is shown in Figure 4.2. Word embeddings are used as word representations (e.g. $\boldsymbol{w}_k$). A Bi-LSTM sequence layer is built on each word over the entire text, computing Bi-LSTM output vectors for each word (e.g. $\boldsymbol{w}^*_k$). The node representation for each time expression or event is the summation of the Bi-LSTM output vectors of all words in the text span (e.g. $\boldsymbol{x}_i$). The pair representation for node i and one of its candidates y'_i is the concatenation of the Bi-LSTM output vectors of these two nodes $\boldsymbol{g}_{i,y'_i} = [\boldsymbol{x}_i, \boldsymbol{x}_{y'_i}]$, which is then sent through a Multi-Layer Perceptron to compute a score for this pair s_{i,y'_i}. Finally all pair scores of the current node i are concatenated into vector $\boldsymbol{c}_i$, and taking $softmax$ on it generates the final distribution $\boldsymbol{o}_i$, which is the probability distribution of each candidate being the parent of node i.

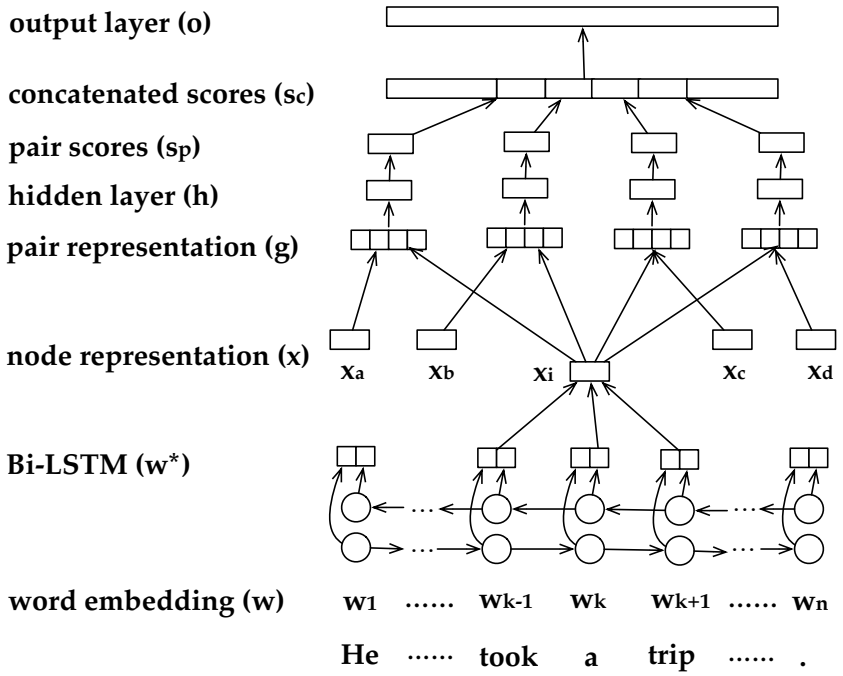


Figure 4.2: Neural Ranking Model Architecture. $\boldsymbol{x_i}$ is the current child node, and $\boldsymbol{x_a}, \boldsymbol{x_b}, \boldsymbol{x_c}, \boldsymbol{x_d}$ are the candidate parent nodes for $\boldsymbol{x_i}$. Arrows from Bi-LSTM layer to $\boldsymbol{x_a}, \boldsymbol{x_b}, \boldsymbol{x_c}, \boldsymbol{x_d}$ are not shown.

Formally, the **Forward Computation** is:

$$\boldsymbol{w}_k^* = BiLSTM(\boldsymbol{w}_k)$$

$$\boldsymbol{x}_i = sum(\boldsymbol{w}_{k-1}^*, \boldsymbol{w}_k^*, \boldsymbol{w}_{k+1}^*)$$

$$\boldsymbol{g}_{i,y_i'} = [\boldsymbol{x}_i, \boldsymbol{x}_{y_i'}]$$

$$\boldsymbol{h}_{i,y_i'} = tanh(\mathbf{W_1} \cdot \boldsymbol{g}_{i,y_i'} + \mathbf{b_1})$$

$$s_{i,y_i'} = \mathbf{W_2} \cdot \boldsymbol{h}_{i,y_i'} + \mathbf{b_2}$$

$$\boldsymbol{c}_i = [s_{i,1}, ..., s_{i,i-1}, s_{i,i+1}, ..., s_{i,i+t}]$$

$$\boldsymbol{o}_i = softmax(\boldsymbol{c}_i)$$

4.5.2 Learning

Let D be the training data set of K texts, N_k the number of nodes in text D_k, and y_i the gold parent for node i. Our neural model is trained to maximize $P(y_1, ..., y_{N_k}|D_k)$ over the whole training set. More specifically, the cost function is defined as follows:

$$\begin{aligned} C &= -log \prod_{k}^{K} P(y_1, ..., y_{N_k}|D_k) \\ &= -log \prod_{k}^{K} \prod_{i}^{N_k} P(y_i|D_k) \\ &= \sum_{k}^{K} \sum_{i}^{N_k} -logP(y_i|D_k) \end{aligned}$$

For each training example, cross-entropy loss is minimized:

$$\begin{aligned} L &= -logP(y_i|D_k) \\ &= -log \frac{exp[s_{i,y_i}]}{\sum_{y'_i} exp[s_{i,y'_i}]} \end{aligned}$$

where s_{i,y'_i} is the score for child-candidate pair (i, y'_i) as described in Section 4.5.1.

4.5.3 Decoding

During decoding, the parser constructs the temporal dependency tree incrementally by identifying the parent node for each event or time expression in textual order. To ensure the output parse is a valid dependency tree, two constraints are applied in the decoding process: (i) there can only be one parent for each node, and (ii) descendants of a node cannot be its parent to avoid cycles.

Candidates violating these constraints are omitted from the ranking process.[1]

4.5.4 Temporal Relation Labeling

The neural model described above generates an unlabeled temporal dependency tree, with each parent being the most salient reference time for the child. However it doesn't model the specific temporal relation (e.g. "before", "overlap") between a parent and a child. We extend this basic architecture to both identify parent-child pairs and predict their temporal relations. In this new model, instead of ranking child-candidate pairs (i, y'_i), we rank child-candidate-relation tuples (i, y'_i, l_k), where l_k is the kth relation in the pre-defined set of possible temporal relation labels L. We compute this ranking by re-defining the pair score s_{i,y'_i}. Here, pair score s_{i,y'_i} is no longer a scaler score but a vector $\boldsymbol{s}_{i,y'_i}$ of size $|L|$, where $\boldsymbol{s}_{i,y'_i}[k]$ is the scaler score for y'_i being the parent of i with temporal relation l_k. Accordingly, the lengths of $\boldsymbol{c}_i$ and $\boldsymbol{o}_i$ are $number\ of\ candidates * |L|$. Finally, the tuple (i, y'_i, l_k) associated with the highest score in $\boldsymbol{o}_i$ predicts that y'_i is the parent for i with temporal relation label l_k.

4.5.5 Variations of the Basic Neural Model

4.5.5.1 Linguistically Enriched Models

A variation of the basic neural model is a model that takes a few linguistic features as input explicitly. In this model, we extend the pair representation $\boldsymbol{g}_{i,y'_i}$ with local features: $\boldsymbol{g}_{i,y'_i} = [\boldsymbol{x}_i, \boldsymbol{x}_{y'_i}, \phi_{i,y'_i}]$.

[1] An alternative decoding approach would be to perform a global search for a Maximum Spanning Tree. However, due to the nature of temporal structures, our greedy decoding process rarely hits the constraints.

Time and event type feature: Stage 1 of the pipeline not only extracts text spans that are time expressions or events, but also labels them with pre-defined categories of different types of time expressions and events. Readers are referred to [Zhang and Xue, 2018c] for the full category list. Through a careful examination of the data, we notice that time expressions or events are selective as to what types of time expression or events can be their parent. In other words, the category of the child time expression or event has a strong indication on which candidate can be its parent. For example, a time expression's parent can only be another time expression or a pre-defined meta node, and can never be an event; and an eventive event's parent is almost certainly another eventive event, and is highly unlikely to be a stative event. Therefore, we include the time expression and event type information predicted by stage 1 in this model as a feature. More formally, we represent a time/event type as a fixed-length embedding $\boldsymbol{t}$, and concatenate it to the pair representation $\boldsymbol{g}_{i,y_i'} = [\boldsymbol{x}_i, \boldsymbol{x}_{y_i'}, \boldsymbol{t}_i, \boldsymbol{t}_{y_i'}]$.

Distance features: Distance information can be useful for predicting the parent of a child. Intuitively, candidates that are closer to the child are more likely to be the actual parent. Through data examination, we also find that a high percentage of nodes have parents in close proximity. Therefore, we include two distance features in this model: the node distance between a candidate and the child $\boldsymbol{nd}_{i,y_i'}$, and whether they are in the same sentence $\boldsymbol{ss}_{i,y_i'}$. One-hot representations are used for both features to represent according conditions listed in Table 4.1.

The final pair representation for our linguistically enriched model is as follows:

$$\boldsymbol{g}_{i,y_i'} = [\boldsymbol{x}_i, \boldsymbol{x}_{y_i'}, \boldsymbol{t}_i, \boldsymbol{t}_{y_i'}, \boldsymbol{nd}_{i,y_i'}, \boldsymbol{ss}_{i,y_i'}]$$

conditions for feature $\boldsymbol{nd}_{i,y'_i}$:
$i.node\_id - y'_i.node\_id = 1$
$i.node\_id - y'_i.node\_id > 1$ and $i.sent\_id = y'_i.sent\_id$
$i.node\_id - y'_i.node\_id > 1$ and $i.sent\_id \neq y'_i.sent\_id$
$i.node\_id - y'_i.node\_id < 1$
conditions for feature $\boldsymbol{ss}_{i,y'_i}$:
$i.sent\_id = y'_i.sent\_id$
$i.sent\_id \neq y'_i.sent\_id$

Table 4.1: Conditions for node distance and same sentence features.

4.5.5.2 Attention Model on Time and Event Representation

In the basic neural model, a straight-forward sum-pooling is used as the multi-word time expression and event representation. However, multi-word event expressions usually have meaning-bearing head words. For example, in the event "took a trip", "trip" is more representative than "took" and "a". Therefore, we add an attention mechanism [Bahdanau et al., 2014] over the Bi-LSTM output vectors in each multi-word expression to learn a task-specific notion of headedness [Lee et al., 2017]:

$$\alpha_t = tanh(\boldsymbol{W} \cdot \boldsymbol{w}^*_t)$$
$$w_{i,t} = \frac{exp[\alpha_t]}{\sum_{k=START(i)}^{END(i)} exp[\alpha_k]}$$
$$\hat{\boldsymbol{x}}_i = \sum_{t=START(i)}^{END(i)} w_{i,t} \cdot \boldsymbol{w}^*_t$$

where $\hat{\boldsymbol{x}}_i$ is a weighted sum of Bi-LSTM output vectors in span i. The weights $w_{i,t}$ are automatically learned. The final pair representation for our attention model is as follows:

$$\boldsymbol{g}_{i,y'_i} = [\boldsymbol{x}_i, \boldsymbol{x}_{y'_i}, \boldsymbol{t}_i, \boldsymbol{t}_{y'_i}, \boldsymbol{nd}_{i,y'_i}, \boldsymbol{ss}_{i,y'_i}, \hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_{y'_i}]$$

This model variation is also beneficial in an end-to-end system, where time expression and event spans are automatically extracted in Stage 1. When extracted spans are not guaranteed correct time expressions and events, an attention layer on a slightly larger context of an extracted span has a better chance of finding representative head words than a sum-pooling layer strictly on words within a event or time expression span.

4.5.5.3 Contextualized Word Embeddings for Word Representation

In our previous neural model variations, BiLSTM output vectors are used as word embeddings, which are trained together with the rest of the neural network on given training data. This may not give optimal word representations in situations where the training data size is limited. Therefore, another model variation we experimented with is to add pre-trained contextualized word embeddings, in addition to BiLSTM output vectors, as word representations. We used pre-trained Chinese BERT embeddings from Google[2], and extended word representation $\boldsymbol{w}_k^*$ as a concatenation of the two embeddings:

$$\boldsymbol{w}_k^* = [BiLSTM(\boldsymbol{w}_k), BERT(\boldsymbol{w}_k)]$$

where $BERT(\boldsymbol{w}_k)$ is the BERT word embedding for word k; and weighted sum of the last 4 layers of BERT with tuned weights are implemented.

[2] https://github.com/google-research/bert/blob/master/multilingual.md

4.6 Experiments

4.6.1 Data

All of our experiments are conducted on the datasets described in [Zhang and Xue, 2018c]. This is a temporal dependency structure corpus in Chinese. It covers two domains: news reports and narrative fairy tales. It consists of 115 news articles sampled from Chinese TempEval2 datasets [Verhagen et al., 2010a] and Chinese Wikipedia News[3], and 120 fairy tale stories sampled from Grimm Fairy Tales[4]. 20% of this corpus, distributed evenly on both domains, are double annotated with high inter-annotator agreements. We use this part of the data as our development and test datasets, and the remaining 80% as our training dataset.

4.6.2 Baseline Systems

We build two baseline systems to compare with our neural model. The first is a simple baseline which links every time expression or event to its immediate previous time expression or event. According to our data, if only position information is considered, the most likely parent for a child is its immediate previous time expression or event. This baseline uses the most common temporal relation edge label in the training datasets, i.e. "overlap" for news data, and "before" for grimm data.

The second baseline is a more competitive baseline for stage 2 in the pipeline. It takes the output of the first stage as input, and uses a similar ranking architecture but with logistic regression classifiers instead of neural classifiers. The purpose of this baseline is to compare our neural models

[3] https://zh.wikinews.org
[4] https://www.grimmstories.com/zh/grimm_tonghua/index

against a traditional statistical model under otherwise similar settings. We conduct robust feature engineering on this logistic regression model to make sure it is a strong benchmark to compete against. Table 4.2 lists the features and feature combinations used in this model.

time type and event type features:
$i.type$ and $y'_i.type$
if $i.type = absolute\ time$ and $y'_i.type = root$
if $i.type = time$ and $y'_i.type = root$
are $i.type$ and $y'_i.type$ time, eventive, or stative
are $i.type$ and $y'_i.type$ root, time, or event
are $i.type$ and $y'_i.type$ root, time, eventive, or stative
if $i.type = y'_i.type = event$ and $\hat{y}.type = state$, for all $\hat{y}$ between i and y'_i
distance features:
if $i.sent\_id = y'_i.sent\_id$
$i.node\_id - y'_i.node\_id$
if $i.node\_id - y'_i.node\_id = 1$
combination features:
if $i.type = state$ and $i.sent\_id \neq y'_i.sent\_id$
if $i.type = state$ and $i.node\_id - y'_i.node\_id = 1$
if $i.type = y'_i.type = event$ and $i.node\_id - y'_i.node\_id = 1$
if $i.type = state$ and $y'_i.type = event$ and $i.node\_id - y'_i.node\_id = 1$ and $i.node\_id\_in\_sent = 1$ and $i.sent\_id \neq 1$
other features:
if i and y'_i are in quotation marks

Table 4.2: Features in the logistic regression system.

4.6.3 Evaluation

We perform two types of evaluations for our systems. First, we evaluate the stages of the pipeline and the entire pipeline, i.e. end-to-end systems where both time expression and event recognition,

as well as temporal dependency structures are automatically predicted. Our models are compared against the two strong baselines described in Section 4.6.2. These evaluations are described in Section 4.6.3.1.

The second evaluation focuses only on the temporal relation structure parsing part of our pipeline (i.e. Stage 2), using gold standard time expression and event spans and labels. Since most previous work on temporal relation identification use gold standard time expression and event spans, this evaluation gives us some sense of how our models perform against models reported in previous work even though a strict comparison is impossible because different data sets are used. These evaluations are described in Section 4.6.3.2.

All neural networks in this chapter are implemented in Python with the DyNet library [Neubig et al., 2017]. The code is publicly available. For Stage 1, all models are trained with Adam optimizer with early stopping and learning rate 0.001. The dimensions of word embeddings, POS tag embeddings, Bi-LSTM output vectors, and MLP hidden layers are tuned on the dev set to 256, 32, 256, and 256 respectively. POS tags in Stage 1 are acquired using the joint POS tagger from [Wang and Xue, 2014]. The tagger is trained on Chinese Treebank 7.0 [Xue et al., 2010]. For Stage 2, the dimensions of word embeddings, time/event type embeddings, Bi-LSTM output vectors, and MLP hidden layers are tuned on the dev set to 32, 16, 32, and 32 respectively. The optimizer is Adam with early stopping and learning rate 0.001.

4.6.3.1 End-to-End System Evaluation

Stage 1: Time and Event Recognition

For Stage 1 in the pipeline, we perform BIO tagging with the full set of time expression and

event types (i.e. a 11-way classification on all extracted spans). Extracted spans will be nodes in the final dependency tree, and time/event types will support features in the next stage. We evaluate Stage 1 performance using 10-fold cross-validation of the entire data set. We use the "exact match" evaluation metrics for BIO sequence labeling tasks, and compute precision, recall, and f-score for each label type.

We first ignore fine-grained time/event types and only evaluate unlabeled span detection and time/event binary classification to show how well our system identify events and time expressions, and how well our system distinguishes time expressions from events. Table 4.3 shows the cross-validation results on these two evaluations. Span detection and event recognition show similar performance on both news and narrative domains. Time expressions have a higher recognition rate than events in news data, which is consistent with the observation that time expressions usually have a more limited vocabulary and more strict lexical patterns. On the other hand, due to the scarcity of time expressions in the Grimm data, time expression recognition in this domain has a very high precision but low recall, which results in a much lower f-score than news. We also put prior TempEval evaluation results in the table for a rough qualitative comparison. Please note that TempEval time expressions and events are of different definitions than times and events in our framework, hence these numbers are not directly comparable.

Labeled evaluation on full set time/event type classification are reported in Table 4.4. Time expressions have higher recognition rates than events on both domains, and dominant event types ("event", "state", etc.) have relatively higher and more stable recognition rates than other types. Event types with very few training instances, such as "modalized event" ($<7\%$), achieve lower and more unstable recognition rates. Other types with less than 2% instances achieve close to 0 recognition f-scores, and are not reported in this table.

evaluated label	news			grimm		
	p	r	f	p	r	f
all span	.81	.74	.78	.83	.74	.78
time	.83	.81	.82	.97	.62	.76
event	.81	.73	.77	.83	.74	.78
TempEval-2 time*	.90	.82	.86	-	-	-
TempEval-2 event*	.92	.85	.88	-	-	-
TempEval-3 time*	.91	.89	.90	-	-	-
TempEval-3 event*	.81	.81	.81	-	-	-

Table 4.3: Stage 1 cross-validation on span detection and binary time/event recognition, with qualitative comparison with TempEval results. (* TempEval results reported here are the best performance for each task on English in each TempEval. TempEval-1 doesn't have time and event detection tasks. Later TempEvals are on clinical domain and relatively less comparable. Both TempEval 2 and 3 are on news domain only.)

time/event type	news	grimm
vague time	.77	.82
concrete absolute	.67	-
concrete relative	.75	-
event	.61	.77
state	.65	.61
completed	.62	.26
modalized	.46	.31

Table 4.4: Stage 1 cross-validation f-scores on full set time/event type recognition.

Stage 2: Temporal Dependency Parsing

For Stage 2 in the pipeline, we conduct experiments on the six systems described above: a simple baseline, a logistic regression baseline, a basic neural model, a linguistically enriched neural model, an attention neural model, and a model with contextualized word embeddings. All models are trained on automatic spans and time/event types generated by Stage 1 using 10-fold cross-validation, with gold standard edges (and edge labels) mapped onto the auto spans. Evaluations

	model	news				grimm			
		unlabeled f		labeled f		unlabeled f		labeled f	
		dev	test	dev	test	dev	test	dev	test
temporal relation parsing with gold spans	Baseline-simple	.64	.68	.47	.43	.78	.79	.39	.39
	Baseline-logistic	.81	.79	.63	.54	.74	.74	.60	.63
	Neural-basic	.78	.75	.67	.57	.72	.74	.60	.63
	Neural-enriched	.80	.78	.67	.59	.76	.77	.63	.65
	Neural-attention	**.83**	.81	76	.70	**.79**	.79	**.66**	**.68**
	Neural-BERT	**.83**	**.83**	**.77**	**.74**	.78	**.80**	**.66**	**.68**
end-to-end systems with auto spans	Baseline-simple	.39	.40	.26	.25	.44	.47	.27	.25
	Baseline-logistic	.36	.34	.24	.22	.43	.49	.33	.37
	Neural-basic	.37	.36	.21	.23	.42	.45	.33	.35
	Neural-enriched	.51	.52	.32	.35	.44	.49	.33	.37
	Neural-attention	.54	**.54**	.36	**.39**	.44	.49	.35	.39
	Neural-BERT	**.61**	.52	**.40**	.38	**.51**	**.51**	**.42**	**.41**

Table 4.5: Stage 2 results (f-scores) with gold spans and timex/event labels (top), and automatic spans and timex/event labels generated by stage 1 (bottom).

in Stage 2 are against gold standard spans and edges, and evaluation metrics are precision, recall, and f-score on $\langle child, parent \rangle$ tuples for unlabeled trees, and $\langle child, relation, parent \rangle$ triples for labeled trees.

Bottom rows in Table 4.5 report the end-to-end performance of our six systems on both domains. On both labeled and unlabeled parsing, our basic neural model with only lexical input performs comparable to the logistic regression model. And our enriched neural model with only three simple linguistic features outperforms both the logistic regression model and the basic neural model on news, improving the performance by more than 10%. However, our models only slightly improve the unlabeled parsing over the simple baseline on narrative Grimm data. This is probably due to (1) it is a very strong baseline to link every node to its immediate previous node, since in an narrative discourse linear temporal sequences are very common; and (2) most events breaking the temporal linearity in a narrative discourse are implicit stative descriptions which are harder to model with only lexical and distance features. Moreover, attention mechanism improves temporal relation la-

beling on both domains, with both gold and automatic time and event spans. Finally, adding BERT contextualized word embeddings helps in certain experimental settings but the improvements are not consistent across the board.

4.6.3.2 Temporal Relation Evaluation

To facilitate comparison with previous work where gold events are used as parser input, we report our results on temporal dependency parsing with gold time expression and event spans in Table 4.5 (top rows). These results are in the same ballpark as what is reported in previous work on temporal relation extraction. The best performance in [Kolomiyets et al., 2012] are 0.84 and 0.65 f-scores for unlabeled and labeled parses, achieved by temporal structure parsers trained and evaluated on narrative children's stories. Our best performing model (Neural-attention) reports 0.81 and 0.70 f-scores on unlabeled and labeled parses respectively, showing similar performance. It is important to note, however, that these two works use different data sets, and are not directly comparable. Finally, parsing accuracy with gold time/event spans as input is substantially higher than that with predicted spans, showing the effects of error propagation.

4.7 Conclusion

In this chapter, we present the first end-to-end neural temporal dependency parser. We evaluate the parser with both gold standard and automatically recognized time expressions and events. In both experimental settings, the parser outperforms two strong baselines and shows competitive results against prior temporal systems.

Our experimental results show that the model performance drops significantly when automatically

predicted event and time expressions are used as input instead of gold standard ones, indicating an error propagation problem. Therefore, in future work we plan to develop joint models that simultaneously extract events and time expressions, and parse their temporal dependency structure.

Chapter 5

Crowdsourcing Temporal Structure Annotations

5.1 Introduction

In Chapter 3 we have shown that, by providing annotators with detailed guidelines and training them in multiple iterations, the TDT representation can be annotated with high inter-annotator agreement by experts. Chapter 4 further shows that a neural ranking model can be successfully trained on the corpus. However, this "traditional" approach to annotation is time-consuming and expensive. The question we want to answer in this chapter is whether TDT annotation can be performed with crowdsourcing methods, an approach that has gained popularity as a means to acquire linguistically annotated data quickly and cost-effectively for NLP research.

Crowdsourcing has been used to annotate data for a wide range of NLP tasks, including question answering, word similarity, text entailment, word sense disambiguation, machine translation, in-

formation extraction, summarization, semantic role labeling, etc. [Snow et al., 2008, Finin et al., 2010, Zaidan and Callison-Burch, 2011, Lloret et al., 2013, Rajpurkar et al., 2018]. The key to acquiring high quality data via crowdsourcing is to make sure that the tasks are intuitive or can be decomposed into intuitive subtasks. In this chapter, we present a preliminary study on crowdsourcing TDT annotations, and show that it is possible to acquire high quality temporal dependency structures through crowdsourcing, and that a temporal dependency parser can be successfully trained on crowdsourced TDTs.

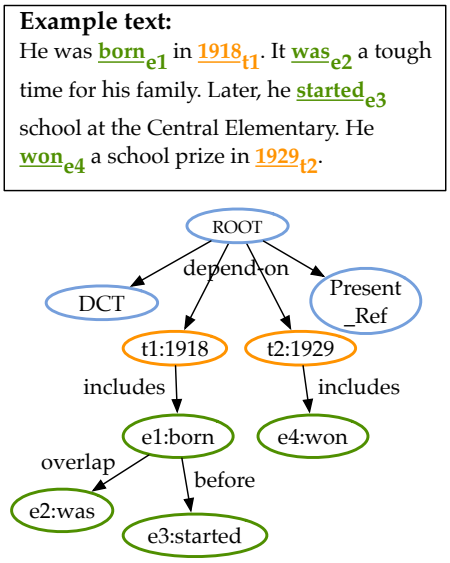


Figure 5.1: Example text and temporal dependency tree. Meta nodes are shown in blue, time expressions in orange, and events in green. TDT also includes meta nodes "Past_Ref," "Future_Ref," and "Atemporal" which are not shown here.

The rest of the chapter is organized as follows. We first explain in detail how we set up this dependency tree crowdsourcing annotation task (§5.2). Annotation experiments (§5.3) show that it is hard to get good inter-annotator agreement when annotating temporal dependency structures in a single step, but when temporal dependency structures are broken into smaller subtasks, high inter-annotator agreement can be achieved. We also experiment with automatic TDT parsers on

this new data, and show that our annotation can support the training of statistical parsers, including an attention-based neural model (§5.4). We discuss related work (§5.5) and conclude with future work (§5.6).

In this preliminary work, we: (1) introduced an effective approach to crowdsource structured temporal annotations, a relatively complex annotation task; (2) built an English temporal dependency tree corpus through crowdsourcing, with high agreements among workers; and (3) experimented with automatic temporal dependency parsers on this new corpus and report competitive results.

5.2 Crowdsourcing Tasks Setup

5.2.1 Data Setup

Our TDT annotations are performed on top of the TimeBank corpus [Pustejovsky et al., 2003b], with time expressions and events already extracted. Following [Zhang and Xue, 2018c], we focus only on events that are matrix verbs in a sentence. In order to extract matrix verbs, we use the gold constituent trees for the part of TimeBank whose gold trees are available in Penn Treebank, and parse the rest of TimeBank with the Berkeley Neural Parser [Kitaev and Klein, 2018], trained on Penn Treebank. All time expressions in TimeBank are kept.

To facilitate quality control in crowdsourcing and agreement evaluation, we distinguish two subsets of the TimeBank dataset: (1) TB-small is a small subset of 10 short Wall Street Journal news documents with 59 matrix verbs. (2) TB-dense consists of the same 36 documents as in the TimeBank-Dense corpus [Cassidy et al., 2014]. It contains 654 matrix verbs. TB-small and TB-dense are annotated by both crowd workers and experts.

5.2.2 Annotation Tasks

We set up two annotation tasks. The first is full temporal dependency tree annotation, where crowd workers need to annotate both the dependency tree structure and the temporal relations between each parent and child. The second is relation-only annotation, where crowd workers are given the gold temporal dependency trees and their job is just to label the temporal relation for each parent-child pair. Although the first task generates both the structure of a TDT and the temporal relations on the edges, we still want to set up the second task as an evaluation analysis to compare temporal relation annotation accuracies in our framework with prior work.

5.2.3 Crowdsourcing Design

For task one, the full temporal dependency tree annotation, in order to simplify the questions/instructions to crowd workers, we split the task of annotating a full dependency tree into (1) finding the "parent" for each individual event, and then (2) deciding the temporal relation between the "parent" and the event. A crowd worker is given a text with a highlighted target event and a list of candidate parent time expresisons and events. The job of the crowd worker is to select one parent from the given list of candidates, and label the temporal relation between the parent and the target event. For task two, relation-only annotation, a crowd worker is presented a text with the target event and its parent highlighted. The job of the worker is to decide the temporal relation between the two.

For quality control, we perform a qualifying test on both annotation tasks. Any crowd worker who wants to work on these tasks needs to complete annotations on TB-small and reach at least 70% accuracy against the expert gold annotation. We also perform a surviving test on the relation-only annotation task. Crowd workers have to maintain at least a cumulative accuracy of 70% for

their annotation. Workers with a lower accuracy will get blocked from the task and all of their annotations will be discarded. Every annotation is completed by at least 3 annotators and the majority vote is the final annotation.

At the end of this chapter, Figure 5.2 and Figure 5.3 give examples on crowdsourcing questions for full structure relation annotation and for relation only annotation respectively.

5.3 Annotation Experiments

We perform the crowdsourcing tasks on the full TimeBank corpus. Crowd worker accuracies against our expert gold annotations on TB-dense and worker agreements on TB-dense and the entire TimeBank data are reported in Table 5.1. The crowd worker accuracy (ACC) is computed against the gold TB-dense annotations, showing how consistent crowd worker annotations are with expert annotations. Worker Agreements With Aggragate (WAWA) [Ning et al., 2018b] represents the agreements among crowd workers themselves, showing how consistent their annotations are with each other. Although the accuracy and agreement for full temporal dependency structure annotation are relatively low, high accuracies and agreements are achieved for both the subtasks of structure annotation and relation-only annotation (above 80%).

	Full	Structure	Relation
TB-Dense ACC	.53	.82	.83
TB-Dense WAWA	.54	.81	.85
TB WAWA	.52	.81	-

Table 5.1: Crowd worker accuracies (ACC) on gold TB-dense and worker agreements (WAWA) on TB-dense and full TimeBank.

Statistics on our corpus and other similar TimeBank-based temporal relation corpora are presented in Table 5.2. As the number of temporal relations is linear to the number of events and time ex-

pressions in a text, fewer temporal relations need to be annotated in our corpus. In comparison, MATRES [Ning et al., 2018b] only annotates verb events in a document while TB-dense annotates a larger number of time expressions and events in a much smaller number of documents. Our corpus retains the full set of TimeBank time expressions and covers comparable number of events as MATRES. We pay $0.01 for each individual annotation and the entire TimeBank TDT annotation cost about $300 in total.

	Docs	Timex	Events	Rels
TimeBank	183	1,414	7,935	6,418
TB-Dense	36	289	1,729	12,715
MATRES	275	-	1,790	13,577
This work	183	1,414	2,691	4,105

Table 5.2: Documents, timex, events, and temporal relation statistics in various temporal corpora.

5.3.1 Crowdsourcing Error Analysis

In order to understand the TDT annotations collected through crowdsourcing more intuitively, we compare them with expert annotations and discuss some error analysis in this section. This analysis is based on the 36 documents from Timebank-Dense that are TDT-annotated both by experts and crowdsourcing workers. Major error types regarding an event's parent include the followings.

Parent Error #1: Overlap Parent Mismatch

In this error, crowdsourcing workers and experts picked different temporal units as the parent for an event. However, the two temporal units actually hold an "overlap" temporal relation and refer to the same temporal location on the timeline. For example, in Text (26) below (event is highlighted in blue, parents are highlighted in orange), the event "helping" happens overlapping this "week" as well as DCT. In this example, experts and crowdsource workers agree on the parent's temporal

location (around this "week" or around DCT), however they picked different timex in the text to represent that temporal location as the specific parent. Text (27) gives another example, the event "learned" overlaps the timex "today". Both indicate the same temporal location on the timeline before which the event "taken" happened. This type of disagreements is settled in the annotation guidelines by specific rules, which are, however, not easily transferable to crowdsourcing workers.

(26): **Text:** On the other hand, it's turning out to be another very bad financial **week** for Asia. The financial assistance from the World Bank and the International Monetary Fund are not **helping**.
Gold: DCT $\xleftarrow{\text{overlap}}$ helping; DCT $\xleftarrow{\text{overlap}}$ week
Crowd: DCT $\xleftarrow{\text{overlap}}$ week $\xleftarrow{\text{overlap}}$ helping

(27): **Text:** Finally **today** we **learned** that the space agency has **taken** a giant leap forward.
Gold: today $\xleftarrow{\text{overlap}}$ learned $\xleftarrow{\text{before}}$ taken
Crowd: today $\xleftarrow{\text{before}}$ taken; today $\xleftarrow{\text{overlap}}$ learned

Parent Error #2: DCT v.s. Close-by Timex Error

For an event that has a timex very close-by both in the text (e.g. in the same sentence or adjacent sentences) and on the timeline (e.g. the event happened right before/after the timex), crowdsource workers tend to pick the timex as the event's parent. However, in many cases the DCT is a more specific parent for the event. For example, in Text (28) below, although "hit a five year low" happened indeed after the "five years", it mostly describes what happened just now around the time of DCT. And in Text (29), although "it's time to reposition" before the "couple of years" of changing, it is actually describing that "it's time ***now (DCT)***".

(28): **Text:** In Singapore, stocks **hit** a **five year** low.

Gold: DCT $\xleftarrow{\text{overlap}}$ hit

Crowd: five year $\xleftarrow{\text{after}}$ hit

(29): **Text:** I think that the mood is fairly gloomy, and I think it's not going to change for **a couple of years**. So for Hong Kong, it's **time** as investment bankers like to say, to reposition.

Gold: DCT $\xleftarrow{\text{overlap}}$ time

Crowd: a couple of years $\xleftarrow{\text{before}}$ time

Parent Error #3: Not Most Specific Parent Error

This error type is a more general form of the last one. In this error, crowdsource workers pick a temporally related parent for an event, but it is not the most specific temporal location the event depends on. For example, in Text (30) below, "you can get seventy percent discounts" in the temporal location "the past three months". It is indeed before DCT, but is also a more specific temporal location than merely "before DCT". In Text (31), "saw an explosion" happened first, then "tries to raise" happened, and then "asks the eastwind pilot" happened last. "tries to raise" is a more specific temporal location than "saw an explosion".

(30): **Text:** But in **the past three months** stocks have plunged, interest rates have soared and the downturn all across Asia means that people are not spending here. Hotels are only thirty percent full. You can **get** seventy percent discounts at the shopping malls.

Gold: the past three months $\xleftarrow{\text{overlap}}$ get

Crowd: DCT $\xleftarrow{\text{before}}$ get

(31): **Text:** We just **saw** an explosion up ahead of us here about sixteen thousand feet or some-

thing like that. It just went down. The controller at Boston center **tries** to raise TWA eight hundred. There is no response. Later, the controller **asks** the eastwind pilot for more details.

Gold: tries $\xleftarrow{\text{after}}$ asks

Crowd: saw $\xleftarrow{\text{after}}$ asks

Parent Error #4: Intentional Event Error

This category of errors is in regard to future intentional events that haven't happened yet (usually realized in the form of verb infinitives). Some intention events are incorrectly identified as parents, some intention events' temporal locations are incorrectly identified. In Text (32) below, "raise" is an intentional event that the controller at Boston center was trying to do. It's unclear whether or not it actually happened after the "trying" event, and is not a valid temporal location to serve as a parent. In Text (33), "demise" was "predicted" but not realized. It doesn't indicate a valid temporal location and cannot serve as a parent.

(32): **Text:** The controller at Boston center **tries** to **raise** TWA eight hundred. TWA eight hundred, if you **hear** center, ident.

Gold: tries $\xleftarrow{\text{after}}$ hear

Crowd: raise $\xleftarrow{\text{after}}$ hear

(33): **Text:** People have **predicted** his **demise** so many times, and the US has tried to hasten it on several occasions. Time and again, he **endures**.

Gold: predicted $\xleftarrow{\text{after}}$ endures

Crowd: demise $\xleftarrow{\text{overlap}}$ endures

Parent Error #5: Quoted Event Error

For quoted events with a timex attached with it, crowdsource workers tend to pick the timex as the parent. However, this is not always necessarily true. For example, in the following text (34), Saddam "said" he will do the event "begin" on timex "Friday". It doesn't mean the event "begin" actually happens on "Friday".

(34): **Text:** In a letter to President Hashemi Rafsanjani of Iran, Saddam **said** he will **begin** withdrawing troops from Iranian territory on **Friday** and release Iranian prisoners of war.
Gold: said $\xleftarrow{\text{after}}$ begin
Crowd: Friday $\xleftarrow{\text{overlap}}$ begin

Parent Error #6: Aspectual Event Error

This category regards aspect events (begin, continue, etc.). Crowdsource workers tend to skip aspect events as possible parents or annotate incorrect temporal relations for them. For example, in the following text (35), crowdsource workers didn't consider "begin" as a valid parent candidate. Please note that, with a finer temporal relation set, "withdrawing" should be "overlap_after" to "begin". However, we are using a coarse temporal relation label set, and such aspectual relations are annotated only as "after" according to our annotation guidelines.

(35): **Text:** In a letter to President Hashemi Rafsanjani of Iran, Saddam **said** he will **begin withdrawing** troops from Iranian territory on Friday and release Iranian prisoners of war.
Gold: begin $\xleftarrow{\text{after}}$ withdrawing
Crowd: said $\xleftarrow{\text{after}}$ withdrawing

Major error types regarding an event' s temporal relation with its parent include the followings.

Relation Error #1: Completed Event Error

For completed events, crowdsource workers tend to consider the temporal location of the "completed" state rather than the temporal location of the "happening" of the event. For example, in Text (36), the state of "having backed out" overlaps with "Now", while the happening of "backing out" was before "Now". And in Text (37), the state of "having become" overlaps with "say (i.e. DCT)", while the happening of "becoming" happened before "say".

(36): **Text:** **Now** with new construction under way, three of his buyers have **backed** out.
Gold: Now $\xleftarrow{\text{before}}$ backed
Crowd: Now $\xleftarrow{\text{overlap}}$ backed

(37): **Text:** Many NASA watchers **say** female astronauts have **become** part of the agency's routine.
Gold: say $\xleftarrow{\text{before}}$ become
Crowd: say $\xleftarrow{\text{overlap}}$ become

Relation Error #2: Modalized Event Error

For modalized events, crowdsource workers tend to consider the temporal location of the "happening" of the event, rather than the temporal location of the "state" the modalized event expresses (e.g. ability, willingness, obligation, etc. to do something). For example, in Text (38), the state of "she can't find buyers" overlaps with the state of her "owning" eight properties. The actual "finding" event hasn't happened yet. And in Text (39), the state of "no one should doubt" overlaps

with him saying the sentence, while the actual "doubting" event might or might not have happened and its temporal location is not of interest here.

(38): **Text:** Pamela **owns** eight condominiums here. She can't **find** buyers.
Gold: owns $\xleftarrow{\text{overlap}}$ find
Crowd: owns $\xleftarrow{\text{after}}$ find

(39): **Text:** "No one should **doubt** our staying power or determination." he **said**.
Gold: said $\xleftarrow{\text{overlap}}$ doubt
Crowd: said $\xleftarrow{\text{after}}$ doubt

There is also a small percentage of other errors, such as close-by timex that's not temporal related to the event incorrectly annotated as the parent, or text with inherent ambiguities. For example, the following sentence (40) has a PP-attachment ambiguity. While experts read it as the naming happens in December, crowdsource workers read it as the mission happens in December and the naming happens beforehand.

(40): **Text:** Air Force Lieutenant Colonel Eileen Collins will be **named** commander of the Space Shuttle Columbia for a mission in **December**.
Gold: December $\xleftarrow{\text{overlap}}$ named
Crowd: December $\xleftarrow{\text{before}}$ named

These error types that crowdsource workers have highly agree with disagreements our expert annotators had during the process of designing our TDT annotation guidelines and through the pilot

annotation experiments on our Chinese TDT corpus. Most of these errors – such as completed events errors, DCT v.s. clost-by timex errors, etc. – can be eliminated with trained annotators through guidelines and rules. However, with untrained crowdsource workers who are not familiar with linguistic concepts such as modalized events, these rules prove to be challenging to implement.

5.4 System Experiments

Experiment 1: Sanity Check and Corpus Baselines

In order to perform a sanity check on our crowdsourced corpus and provide baseline results for future bench-marking, the first experiment we conducted was applying our state-of-the-art attention-based neural temporal dependency parser [Zhang and Xue, 2018a][1] on this new corpus, including crowdsourced and expert annotated data.

Our training data consists of two parts concatenated together. The first part is the crowdsourced temporal dependency annotations over the TimeBank documents (excluding documents that are in the dev and test sets in the TimeBank-Dense corpus[2]). The second part is our expert-annotated TDTs on the TimeBank-Dense training set documents. The parser is tuned and evaluated on our expert TDT annotations on the TimeBank-Dense dev and test sets, respectively. This neural model is the same with the Neural-Attention model as described in Chapter 4. It represents words with bi-LSTM vectors and uses an attention-based machenism to represent multi-word time expressions and events.

[1] `https://github.com/yuchenz/tdp_ranking`

[2] Standard TimeBank-Dense train/dev/test split can be found in [Cassidy et al., 2014].

We also experiment with two baseline parsers from [Zhang and Xue, 2018a]: (1) a simple baseline that takes an event's immediate previous time expresion or event as its parent and assigns the majority "overlap" as the temporal relation between them; and (2) a logistic regression model that represents time expressions and events with their time/event type features, lexical features, and distance features.

The first three rows in Table 5.3 show the performance of these systems on our data with both the large crowdsourced corpus and the small expert-annotated corpus. "Zhang-2018 Simple" and "Zhang-2018 Neural" rows are the performance of the simple baseline and their best neural system on expert-annotated Chinese news data, as reported in [Zhang and Xue, 2018a]. Comparing the simple baseline performance on the two data sets, we can tell that Chinese news data set has a higher proportion of linear *overlap* relations and thus a higher majority baseline than English.

Model	**Struture -only F**		**Structure + Relation F**	
	dev	**test**	**dev**	**test**
Simple Baseline	.43	.42	.15	.18
LogReg Baseline	.64	.70	.36	.39
Neural Model	**.75**	**.79**	**.53**	**.60**
Zhang-2018 Simple	.64	.68	.47	.43
Zhang-2018 Neural	.83	.81	.76	.70

Table 5.3: Parsing results of the simple baseline, logistic regression baseline, and the neural temporal dependency model.

Comparing the neural model performance on the two languages, we can see that the off-the-shelf neural parser performs comparably on the two languages even though the Chinese data sets are annotated with carefully trained annotators. The performance difference on full structure + relation parsing is greater (0.60 v.s. 0.70 f-scores on test set). This is likely due to error propagation through crowdsourcing in a two-staged annotation setup. On the other hand, comparable structure-only parsing performance (0.79 v.s. 0.81 f-scores on test set) shows that our crowdsourced data provide

consistent temporal information that can be learned with statistical models.

Comparisons between the logistic regression baseline and the neural model show that the neural model adapts better to new data sets (and a different language) than the logistic regression model with manually-crafted language-specific features.

Experiment 2: Annotation Quality Check

In order to check the quality of our new corpus from a system's perspective and how effective the approach of crowdsourcing is for collecting large temporal dependency tree data for system training, we performed this experiment for a comparison between TDT parsers trained on gold data V.S. TDT parsers trained on crowdsourced data.

Models are trained on three different data settings. For the first data setting, we used the standard Timebank-Dense data split for training, dev, and test sets, and used only expert-annotated TDT annotations. In other words, systems are trained and tuned with small gold training and dev data, and evaluated against gold test data. For the second data setting, we split our crowdsourced Timebank TDT annotations into training, dev, and test set, using the same documents as in Timebank-Dense dev set for dev, and the same documents as in Timebank-Dense test set for test. The rest documents are the training set. Therefore, systems are trained and tuned with large crowdsourced data, and evaluated against gold test data. For the third data setting, we used crowdsourced TDT annotations on Timebank-Dense only, with the same train/dev/test data split. The purpose of this experiment is to have a direct comparison on how well parsers can be trained using the same amount of expert data V.S. crowdsourced data.

Experimental results for both logistic regression and neural model are illustrated in Table 5.4. It's evidently shown that across the board, a large crowdsourced training/dev dataset is very helpful on

improving parser performances over a small gold training/dev dataset. Moreover, even only on a small dataset, TDT parsers can be trained to perform basically as well on crowdsourced annotations as on expert annotations.

Model	Training / Dev Data	Structure + Relation F		Structure -only F	
		dev	test	dev	test
Baseline	-	.15	.18	.43	.42
LogReg	Gold, Small	.28	.34	.46	.49
	Crowdsourced, Small	.28	.33	.45	.51
	Crowdsourced, Large	.30	.35	.50	.53
	Crowd Large + Gold Small	.36	.39	.64	.70
Neural	Gold, Small	.42	.45	.60	.60
	Crowdsourced, Small	.41	.47	.60	.59
	Crowdsourced, Large	.49	.53	.66	.69
	Crowd Large + Gold Small	.53	.60	.75	.79

Table 5.4: Comparison between TDT parsers trained on gold data V.S. TDT parsers trained on crowdsourced data.

5.5 Related Work

Although crowdsourcing is widely used in other NLP tasks, there have been only a few temporal relation annotation tasks via crowdsourcing. The first attempt on crowdsourcing temporal relation annotations is described in [Snow et al., 2008]. They selected a restricted subset of verb events from TimeBank and performed strict before/after temporal relation annotation through crowdsourcing. They reported high agreements showing that simple temporal relations are crowdsourceable. [Ng and Kan, 2012] adopts the TimeBank temporal representation, and crowdsourced temporal annotations on news articles crawled from news websites. Their experiments show that the large crowdsourced data improved classifier performance significantly. However, both of these works

focused on pair-wise temporal relations and didn't experiment with crowdsourcing more complex temporal structures.

[Ning et al., 2018b] proposes a "multi-axis" representation of temporal relations in a text, and annotates this representation on the TempEval-3 corpus through crowdsourcing. They argue that events need to be annotated on different "axes" according to their eventuality types, and for events on the same axis, pair-wise temporal relations are annotated. Their annotation task is broken down to two smaller subtasks too. In the first subtask, crowd workers annotate whether an event is on a given axis. In the second subtask, crowd workers annotate the temporal relations between pairs of events on the same axis. The main differences between their work and ours are as follows. First, they only model events, excluding time expressions which are important temporal units in text too. Second, our temporal dependency tree representation is very different from their multi-axis temporal representation, which requires different crowdsourcing task designs. In their first subtask, crowd workers need to distinguish different eventuality types, while our annotation experiments show that crowd workers can also consistently recognize "parents" as defined in [Zhang and Xue, 2018c] for given events.

5.6 Conclusion and Future Work

In this chapter, we introduce a preliminary study on a crowdsourcing approach for acquiring annotations on a relatively complex NLP concept – temporal dependency structures. We build the first English temporal dependency tree corpus on top of TimeBank through high quality crowdsourcing. Our system experiments show that competitive temporal dependency parsers can be trained on our newly collected data. Errors and issues with this preliminary crowdsourcing approach are discussed, showing promising future directions of research.

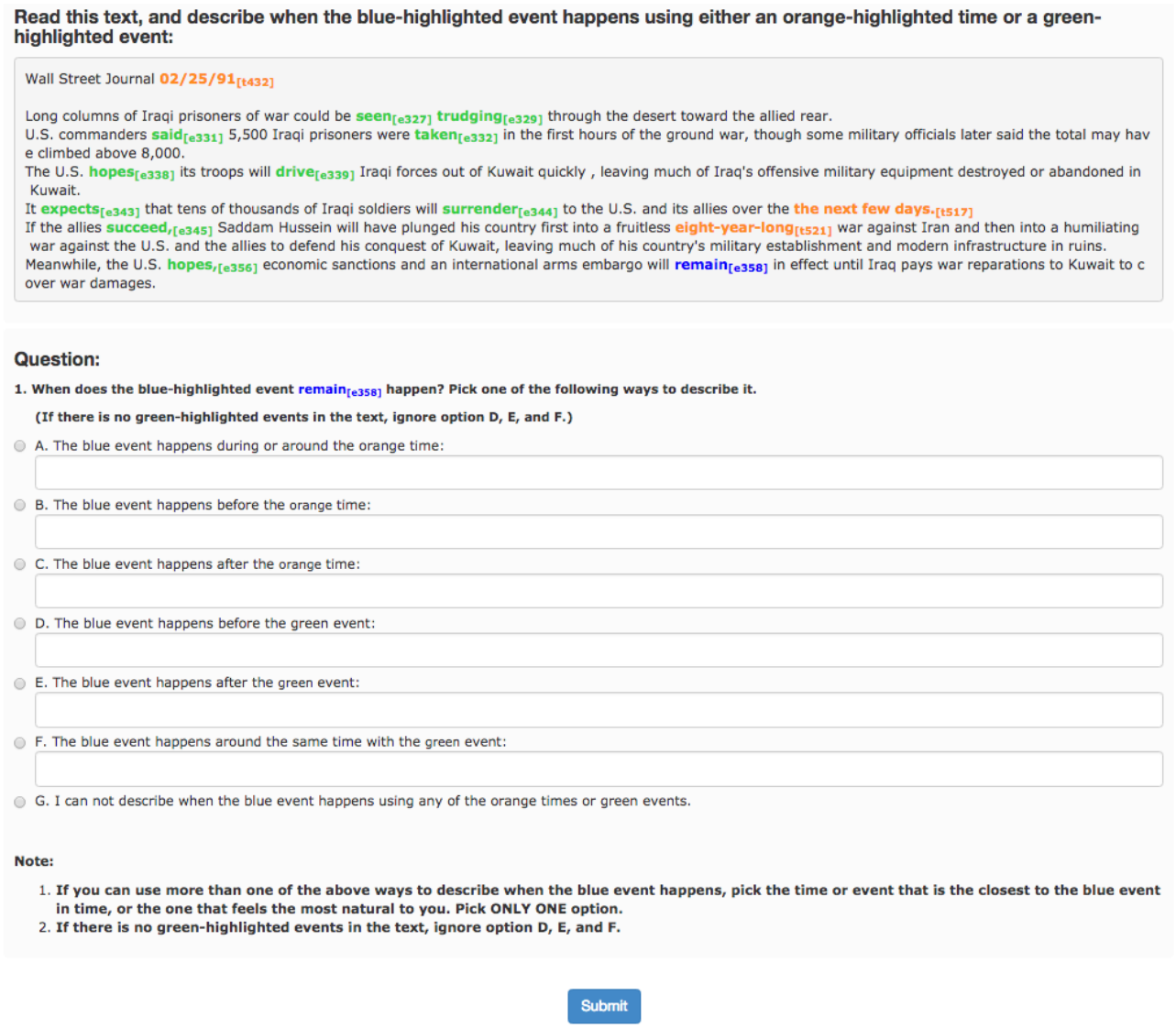


Figure 5.2: Example crowdsourcing question for full structure and relation annotation. Crowdsource workers will read this passage, recognizing the event in question (blue), all time expressions (orange), and candidate event parents (green). Then they will consider when the blue event happens, and with which time expression or candidate parent event they can describe it the best. For example, if a crowdsource worker decides that "remain" happens after "hopes", then he will pick the option (E.) and copy "hopes,[e356]" into the blank text box under option E.

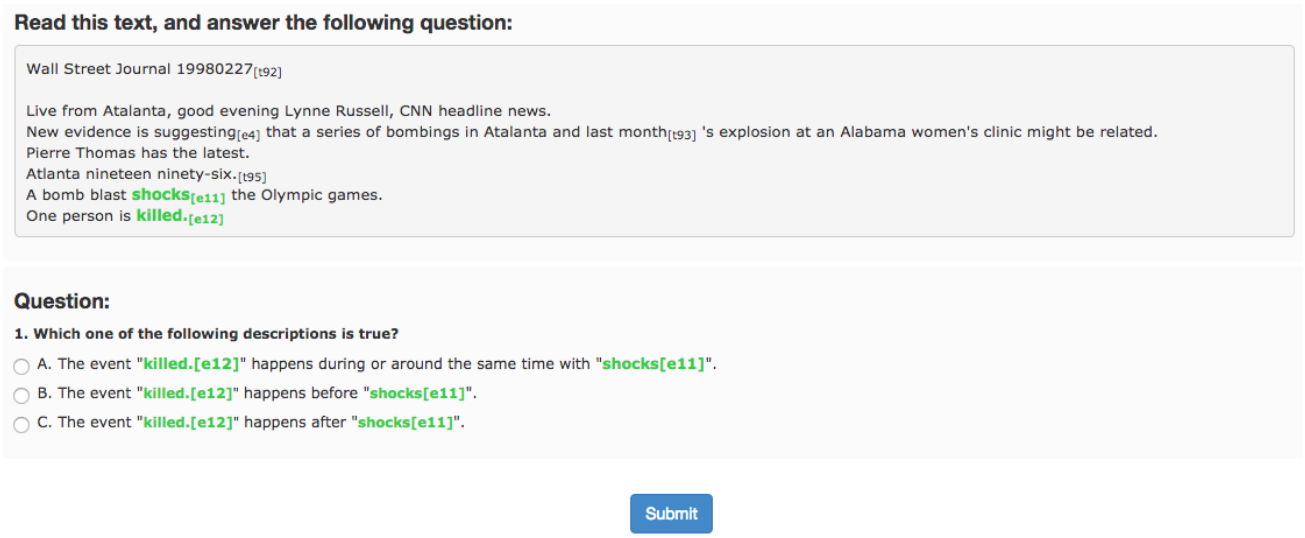


Figure 5.3: Example crowdsourcing question for relation only annotation. Crowdsource workers will read this passage, recognizing the two events in question. Then they will consider the temporal relation between the two events, and pick the according option.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

In this thesis, we present research around various aspects of temporal information modeling: from temporal representation, to temporal structure data collection, then to automatic temporal parsers.

To overcome the issues of redundancy and conflicts in pair-wise temporal relation representations without introducing computationally expensive global constraints, we designed a novel representation to model temporal information in text – the Temporal Dependency Tree (TDT) Structure. We show that this structure is linguistically intuitive, and is amenable to computational modeling. As a proof-of-concept and resource for further research, we built a TDT corpus of 235 documents in Chinese, covering two domains: news and narratives. High and stable inter-annotator agreements in our annotation experiments provide further evidence supporting this structured interpretation of temporal relations. This corpus is publicly available for future research on temporal relation analysis, story timeline construction, as well as numerous other applications.

To enable computers to automatically learn and parse TDT structures, we built the first end-to-end neural temporal dependency parser. This parser was evaluated with both gold standard and automatically recognized time expressions and events. In both experimental settings, the parser outperforms two strong baselines and shows competitive results against prior temporal systems.

In order to collect TDT data more effectively and efficiently, we proposed a preliminary crowdsourcing approach to acquire TDT annotations. Since TDT structure is a very complex structure for crowdsourcing workers, this approach was specially designed to simplify complicated linguistic concepts in TDT and the task in general. This approach was evaluated by crowdsourcing annotation experiments on English Timebank corpus. We show that high quality TDT structure annotations can be collected through our specially-designed crowdsourcing approach. To build English TDT resource, we collected English TDT annotations on top of the Timebank corpus (183 documents in total) using this crowdsourcing approach. Finally, we extended our neural TDT parser to the English TDT corpus. System experiments show that our parser can be easily applied to English TDT parsing without much modification. Although these results are still preliminary, they show promising directions of future research.

Temporal information modeling, including but not limited to temporal representation, temporal corpora, and temporal parsing, is a very important task to natural language understanding. However, it's still a growing field of research and we still have a long way to go in order to put such technique to practice. We summarize the promising future directions related to temporal information modeling in the rest of this chapter, in the hope that these ideas will inspire researchers in this field and lead to further improvements.

6.2 Future Directions

6.2.1 Chinese Temporal Machine Reading Comprehension with TDT

Machine Reading Comprehension tasks have attracted a large amount of research interest in recent years. From cloze-style MRC tasks [Cui et al., 2016, Cui et al., 2017, Zheng et al., 2019], to span-extraction MRC tasks [Cui et al., 2018, Shao et al., 2018, Yao et al., 2019], and to multi-document open-domain MRC tasks [He et al., 2018, Li et al., 2016], researchers have been interested in building better and larger MRC corpora in English, Chinese and other languages, as well as developing and improving better MRC systems. Since Temporal Dependency Tree structure models events and temporal relations in a computational efficient representation, we are curious to see if it helps Machine Reading Comprehension (MRC) tasks regarding temporal-related questions.

In this work, we propose to apply TDT structure onto single-document span-extraction MRC, focusing on temporal-related questions only. From existing Chinese MRC datasets, we collected 5,637 temporal-related (context, question, answer) tuples (see Table 6.1). Questions asking about when something happens (e.g. questions containing "when") are filtered out as temporal-related questions.

Dataset	# Temporal Qs	# Qs
CMRC-2018 [Cui et al., 2018]	798	14k
DRCD [Shao et al., 2018]	1,537	33k
SMART [Yao et al., 2019]	3,302	39k
Total	5,637	86k

Table 6.1: Statistics on temporal questions in existing MRC datasets.

With proper data preprocessing (word segmentation, event and time expression extraction, and TDT parsing), we propose to build a system that (1) matches the event mentioned in a question to an

event in the context through data-driven or heuristic approaches, and (2) finds the time expression for the event through a heuristic-based temporal reasoning process on TDT. To evaluate this system, we will compare its performance with existing (already trained) MRC systems' performance on a set of temporal-related questions.

6.2.2 Chinese Temporal MRC Dataset Construction with TDT

A preliminary search on temporal-related MRC questions shows that although there are abundant corpora on MRC tasks, the amount of temporal-related questions is very limited. Within the a few span-extraction MRC datasets we have looked at, temporal-related questions only take up around 6.5% of the entire data (see Table 6.1). This small data size could potentially limit the research and development of temporal-oriented MRC systems. Therefore, we propose to construct a Chinese Temporal MRC dataset using existing annotations on our TDT corpus. We propose to (1) build a heuristic-based tool that generates full event descriptions based on an event anchor on TDT (either extractive or abstractive descriptions, probably with the help of syntactic parsing trees); (2) generate temporal-related questions and answers regarding these events using heuristic methods; and (3) provide baseline results by applying existing MRC systems (already trained) on this dataset.

6.2.3 Life Events / Historical Events Timeline Construction with TDT

Timeline Summarization or Storyline Construction is the task of organizing crucial milestones of a news story in a temporal order. Most prior research focused on multi-document summarization, and aimed at summarizing a large number of short news reports on the same topic, to construct a

timeline of the news story. In these tasks, the Document Creation Times are usually mainly used as the timestamps on the timeline. There have been little single-document timeline construction tasks. However, for articles that are temporally-organized in nature, such as a person's life events descriptions (e.g. Einstein's "Early Life" descriptions on Wikipedia) or historical events descriptions (e.g. descriptions of the Battle of Midway on Wikipedia), a timeline list of events is a concise summarization of the article, a clear representation of the historical time period, and can potentially help human comprehension of the person or the history.

In this work, we propose to apply TDT structure onto temporally-organized articles, such personal life biographies, historical event descriptions, etc., to construct a timeline of major events happened in a certain historical event, or a person's certain period of lifetime. We propose to (1) dump relevant data from Wikipedia (i.e. articles on celebrity life descriptions, or historical event descriptions), and manually select a small set of articles; (2) preprocess data with word segmentation, time expression and event extraction, and TDT parsing; and (3) build a heuristic-based system that generates full event descriptions based on an event anchor on TDT (either extractive or abstractive descriptions, probably needs the help of syntactic parsing trees), and converts a TDT into a timeline list of events.

6.3 Other Future Directions

Our experimental results of the neural TDT parser on the Chinese TDT corpus show that the model performance drops significantly when automatically predicted event and time expressions are used as input instead of gold standard ones, indicating an issue of error propagation. A joint model that simultaneously extracts events and time expressions, as well as parses their temporal dependency structures will alleviate the error propagation problem, and indicates a possible research direction.

Parsing experiments on our Chinese TDT corpus show that our parsers perform much better on the news data than the narrative stories. Since our parsers are trained on two domains separately, Domain Adaptation techniques are potentially useful here for leveraging data on multiple domains to train better parsers for narrative stories, and for building cross-domain TDT parsers.

Lastly, since our expert-annotated Chinese TDT data and crowdsourced English TDT data are relatively small corpora, future directions also include more TDT data crowdsourcing. Larger TDT corpora will support further TDT parser development and improvements.

Appendix

Appendix A

A.1 Chinese Temporal Dependency Tree Annotation Guidelines

A.1.1 Time Expression Recognition

The first pass of annotation is to mark out spans that are time expressions (timex), which are the backbones of the final temporal dependency parses. For Grimm and Wikinews data, annotation from scratch is needed. For TempEval2 data, its originally annotated timex are adopted first, then small modifications are applied and some missing timex are added, too.

Some rules to help marking out timex spans are:

1. Two or more timex that are next to (or very close to) each other and express one temporal location that is the reference time for nearby events should be merged into one span and marked as one timex.

- 去年下半年月月逆差
- 1997年全年的累计顺差额降到微不足道的数目
- 从1994年11月18日破土动工到锥炉点火成功
- 今年一至十一月份
- 今明两年
- “七五”期间（一九八六至一九九零年）
- 二十世纪初期

2. Timex that are durations with a temporal aspect marker should be merged with the marker to form a complete temporal location.

   - 五年前
   - 中国已经确定未来五年高技术研究重点

3. A name that is specifically given to refer to a period of time on the timeline should be marked as a timex.

   - “八五”期间
   - 战国时期

A.1.2 Time Expression Classification

The second pass of annotation is to give every timex a label, making some characteristics about the timex explicit, which will be helpful for downstream annotations.

In this labeling task, we make a distinction between timex that are temporal locations and the ones that are not. We define a timex that can be located on the timeline, and express a starting and an ending temporal boundary as a temporal location. For example, "1997年", "一至十一月份", "目前", "从前", "二十世纪初期", "20分钟之后" are all temporal locations, but "每秒", "连年", "初期患者", "他标注一篇大概需要20分钟" are all NOT temporal locations. As shown here, two timex with the same lexical words can express both a temporal location and a non-temporal location, depending on their context. When considering whether a timex is a temporal location, look at both the timex itself and its context, and consider the following rules to help you make the distinction. The final goal of this annotation project is to find a reference time for each event, so we only care about the timex that can temporally locate to a span on the timeline. Therefore, we give all non-temporal location timex the label "Timex-Ignore", not considering them as valid reference times for events.

Some rules to help making the distinction between temporal locations and non-temporal locations are:

1. A timex that expresses the concept of a duration of time that can not be anchored to the timeline is labeled "Timex-Ignore".
    - 仅用了十三个月，工期比一期缩短了十八个月，比世界建设最快的同类项目还提前了五个月
    - 为期十五年的"八六三" 计划
    - 每年都有几次历时半月以上
    - 长期发展慢的电子类产品发展加速
2. A timex that expresses a unit time for measurement that can not be anchored to the timeline

is labeled “Timex-Ignore”.

- 每年都有几次历时半月以上
- 每月只需交纳极低的租金
- 年产12万吨聚氯乙烯, 年产量超过设计能力百分之七十八
- 我们自己开发的计算机每秒计算速度
- 扭转了对外贸易连年逆差的形势

3. A timex that’s an ordinal is labeled “Timex-Ignore”.

   - 并连续第三年实现顺差

4. A timex that expresses a phase of a generic process is labeled “Timex-Ignore”.

   - 晚期乙肝疾病往往会转化为肝癌

5. Timex that refer to events as temporal locations are “Timex-Ignore”. These events will be marked, and surrounding events will be temporally located based on these events.

   - 建国初期，重庆是中共中央西南局和西南军政委员会所在地，为中央直辖市。
   - 战后初期，苏、美、英、法四大国据据“ 雅尔塔协议” 划分了势力范围，并分区占领了德国及其首都。

6. Timex such as “当时”, “那时”, “不久”, “期间”, etc. that are referring to other events are “Timex-Ignore”. The temporal locations between these events will be represented by their relations between each other, rather than their relations with these timex.

   - 战后初期，苏、美、英、法四大国据据“ 雅尔塔协议” 划分了势力范围，并分区占领了德国及其首都。当时德国的最高决策当局是“ 盟国管制委员会” 。德国此

后长期处于无权和被分裂的状态。不久，美苏两国在欧洲的较量使原来的反法西斯联盟发生了分裂，德国也一分为二。在西边，美国组织北约，扶植西德，以“ 遏制” 苏联；在东边，苏联组建华约集团，把东德建成对付西方的桥头堡。分裂的德国成了东西方冷战对峙的前线。从那时起，欧洲政局无一日安宁，危机重重。

For timex that are temporal locations, we make another two distinctions. The first distinction is between concrete timex and vague timex. Concrete timex are timex that express a specific temporal location. For example, “1997年”, “六十年代中期”, “21日” are all concrete timex. Their starting and ending temporal boundaries on the timeline can be determined. We consider both exact boundaries and loose boundaries as valid temporal location boundaries. For example, “1997年” has exact starting and ending temporal boundaries as 1997-01-01:00:00:00 and 1997-12-31:24:59:59, while “六十年代中期” has loose temporal boundaries as, depending on different people’s interpretation of “中期”, maybe 1963 to 1967. Usually a concrete timex that has exact temporal boundaries plus a timex indicating the sub-part in it is considered a concrete timex with loose boundaries. For example, “初”, “初期”, “月末”, “年底”, “中期” can all be attached to the end of a concrete timex forming another concrete timex. Vague timex are timex that express the concept of (or a period in) general past, general present, or general future, without specific temporal location boundaries. Some examples are “目前”, “近几年”, “从前”, “有一天”.

The second distinction is made only among concrete timex. It’s between absolute concrete timex and relative concrete timex. Absolute timex are timex that contain all information needed to be located on the timeline. For example, “1997年”, “六十年代中期”, “‘八五’期间” are all absolute timex. One can easily find their starting and ending temporal boundaries on the timeline without help from other spans of text. Relative timex are timex that need the help from other timex to

interpret their temporal locations. For example, “今年”, “一至十一月份”, “过去三年” are all relative timex. “今年”’s temporal location is dependent on the Document Creation Time (DCT) which is usually a metadata of the yyyy-mm-dd format. The timex itself “今年” only expresses the meaning “in the same year with DCT”. If, for example, the year of DCT is 1997, we can then interpret “今年” as 1997 on the timeline. Similarly, “一至十一月份” expresses “January to November in the year of DCT”, and “过去三年” expresses “the past three years before the year of DCT”. Timex other than DCT can be the reference timex as well. For example, in the following discourse,

- 约恩·乌松，丹麦籍的著名建筑设计师，于11月29日在睡梦中病逝，享年90岁。乌松近年来一直被心脏病困扰，今年还做过几次手术，26日曾经心脏病发。

“11月29日”, depending on DCT (which for example is 2008-12-01), will be interpreted as 2008-11-29, and “26日”, depending on “11月29日”, will then be interpreted as 2008-11-26.

These two distinctions help with the next pass of annotation, the annotation of timex parses, in (1) determining whether a timex needs a reference time (relative timex do and absolute timex don’t), and (2) whether the timex needs a general past/present/future reference time, or a specific timex or DCT reference time (concrete timex need a specific one and vague timex need a general one).

For future work, these two distinctions will also help with timex normalization. Vague timex’s normalizations are merely their general reference time, concrete absolute timex’s normalizations can be computed using their lexical words, and concrete relative timex’s normalizations can be computed using their lexical words together with their reference time’s normalization.

In other words, this pass of annotation is a four-way classification task. The four categories are: vague timex, concrete absolute timex, concrete relative timex, and timex that are not temporal

locations and need to be ignored. The four labels we use are:

- Timex-ConcreteAbsolute
- Timex-ConcreteRelative
- Timex-VagueRelative
- Timex-Ignore

Here we consider all vague timex as relative because they depend on either a general past/present/future reference time or another vague timex to be located on the timeline. Examples for vague timex depending on another vague timex are most seen in narrative stories. For example, in the following discourse,

- 从前有一个家境贫穷，但是心地善良的小姑娘和母亲过着孤苦伶丁的生活，她们总是吃不饱肚子。有一天小姑娘走进森林，遇到了一位老婆婆。

"从前" is a vague timex that depends on general past, and "有一天", depending on "从前", will be interpreted as a temporal location that spans one day's length on the timeline in the general past section.

Some examples for different time expression categories are:

A.1.3 Time Expression Reference Time Resolution

The third pass of the annotation is to find the reference time for each relative timex. Vague timex's reference times can be general past/present/future, or another vague timex. Concrete relative timex's reference timex can be DCT, or another concrete timex. No further annotations are

	Absolute	Relative
Concrete	1997年, “八五”期间, 1995年底*, 六十年代中期*	这一年, 今年, 上年, 过去三年, 去年7月份, 去年下半年, 今年一至十一月份, 上半年, 本世纪末*, 1月20日, 一至十一月份, 21日, 九月初*, 星期三
Vague	-	日前, 目前, 近几年, 现, 同时, 从前, 有一天

Table A.1: Some examples for different timex types.

needed on concrete absolute timex and ignored timex. For each reference time/timex pair, annotate a link from the reference time to the timex, representing the relation of the timex depending on the reference time to be located temporally, so no link label is needed.

Following TimeML, we use symbols PAST_REF, PRESENT_REF, and FUTURE_REF to denote general past/present/future. Some example vague timex of these reference times are:

- PAST_REF: 日前, 过去,
- PRESENT_REF: 目前, 现, 近年, 近几年, 近日,
- FUTURE_REF: 本世纪末, 将来,

After annotating the reference time for each relative timex, an automatic process will take place to build the final timex parse, which links DCT, general past/present/future, and all concrete absolute timex directly to a ROOT node, forming a complete tree structure which can be used in future work for timex normalization and timeline structure building.

An example of the final parse is illustrated in the following figure. (We also have an ATEMPORAL node linked as a child of the ROOT. It will be used in the last pass of annotation, event parsing.)

The possible reference time/node for different type of timex is summarized in the following table.

	Absolute	Relative
Concrete	ROOT	DCT, or another Concrete Timex
Vague	-	PAST_REF, PRESENT_REF, FUTURE_REF, or another vague timex

Table A.2: Possible reference times or nodes for different types of timex.

A.1.4 Event Recognition

The fourth pass of annotation is to mark out spans that are events. In this stage, two decisions need to be made during annotation: (1) whether something should be considered as an event; and (2) what exact span of words should be marked to designate the event.

Adapting from TimeML event annotation guidelines, we consider occurrences, actions, processes, or event states which deserve a place upon a timeline as events. However, in this task, we only work with a subset of the events defined as markables in TimeML – events that are the main predicates in a sentence and a limited set of subordinate clauses. These events' syntactic realizations are mostly verbs with a few exceptions of nominalizations, nouns, and adjectives described more later. The reasons that we select this subset of events are that (1) main predicates advance the temporal progress of a narrative or reporting discourse, and their temporal relations are dependent to one another, forming the temporal structure we are trying to capture in this task; and (2) other events (e.g. events in relative clauses) are merely mentions of a temporal location that are independent of other events or timex in the discourse, thusly independent to the structure we are trying to build, and marking those events and locating them on a timeline is out of the scope of this work.

Decision 1: What are considered as events?

More specifically, the following predicates are considered markable events in our task:

1. The main predicates of a sentence/independent clause

   One sentence usually has one main predicate. If there are two main predicates joined by coordinating conjunctions, both are markable events.

   - 由中国自主设计建设、达到当今世界先进技术水平的安阳彩色显像管玻壳有限公司二期工程，今天建成(event)。
   - 能否把我们自己的高技术及其产业搞上去，关系(event)到中国现代化建设事业的成败，关系(event)到中华民族的兴衰。
   - 大力发展高技术，尽快形成我国强大的民族高技术产业，是(event)当前中国科技界和经济界面临的迫切任务。

2. Predicates in adverbial clauses (serving as time)

   - 实行(event) 保证金台帐制度后，海关对正常开展加工贸易的企业不再征收与进口料件税款等值的风险保证金，只是在银行设立(event) 台帐时收取一百元的手续费，因而将减轻企业的实际经济负担。

3. Nouns in adverbials serving as temporal locations

   - 建国(event) 初期，重庆是中共中央西南局和西南军政委员会所在地，为中央直辖市。
   - 战(event) 后初期，苏、美、英、法四大国据据“ 雅尔塔协议” 划分了势力范围，并分区占领了德国及其首都。

On the other hand, the following predicates are NOT considered markable events in our task:

1. Predicates in noun clauses (serving as subjects and objects)

- 今年国家实行外经贸三大政策调整包括(event) 调低(NOT event) 出口退税率、对加工贸易进口料件实行(NOT event) 银行保证金台帐制度和取消(NOT event) 进口设备免税优惠。
- 能否把我们自己的高技术及其产业搞(NOT event)上去，关系到中国现代化建设事业的成败，关系到中华民族的兴衰。大力发展(NOT event)高技术，尽快形成(NOT event)我国强大的民族高技术产业，是当前中国科技界和经济界面临的迫切任务。

Two exceptions for predicates in noun clauses are:

First, when the noun clause is serving as the object of a reporting verb, such as "指出", "说", "表明", the main predicates in the noun clause are considered markable events.

- 广东省外经贸委有关负责人指出(event) ，实行加工贸易台帐制度是(event) 为了完善对加工贸易的监管，堵塞管理漏洞，防止国家税收流失，促进加工贸易的健康发展。

Second, when the noun clause is serving as the subject or object of the main verb "是" whose purpose is only to emphasize the noun clause, the main predicates in the noun clause are considered markable events. In such cases, the main verb "是" is not considered as an event.

- 尤为值得一提的是(NOT event) ，" 八五" 期间中国的对外开放已形成(event) 了从沿海、沿江向内陆边远地区梯次推进的格局，以往经济相对落后的内陆地区如今也掀起(event) 了开放的热潮。

2. Predicates in adverbial clauses (serving as purpose, reason, condition, place) to do: consider adverbial clauses serving as concession, results, comparison, and manner

- 广东省外经贸委有关负责人指出(event)，实行加工贸易台帐制度是(event) 为了完善(NOT event) 对加工贸易的监管，堵塞(NOT event) 管理漏洞，防止(NOT event) 国家税收流失，促进(NOT event) 加工贸易的健康发展。
- 要在2000年实现(NOT event) 人均国内生产总值五千美元的目标, …

3. Predicates in relative clauses

   - 由中国自主设计建设(NOT event)、达到(NOT event) 当今世界先进技术水平的安阳彩色显像管玻壳有限公司二期工程，今天建成。

Decision 2: What exact span of words to mark?

Generally the verb of a predicate is marked to designate the event. A few exceptions are as follows:

1. In Chinese, there are cases where the verb in a predicate is dropped. For such cases, simply mark the object as the event. 目前，全省从事加工贸易人数五百多万, 仅“三来一补”的企业就达三万多家，从业人员二百多万。

2. For negated verbs, mark the negation and the main verb together as one event. 跨国公司在山东投资势头不减。实行保证金台帐制度后，海关对正常开展加工贸易的企业不再征收与进口料件税款等值的风险保证金，只是在银行设立台帐时收取一百元的手续费，因而将减轻企业的实际经济负担。

3. For modalized verbs, mark the modalized verb only as the event. 投产后，它既可生产与彩电配套的彩色显像管玻壳，…

4. For cases with an aspectual verb followed by a main verb, mark the aspectual verb and the main verb as separate events.

5. For predicates that are expressed as "light verb + nominalized/main verb", mark the nominalized/main verb only. The light verb is not considered as an event.

   - 有+ nominalized/main verb:
     - 有上升
     - 有二十个重点项目竣工投产
   - 使+ main verb:
     - 使四大类产品形成了规模经济
   - 进行+ nominalized event:
     - 进行企业改造和发展
   - Other verbs are not considered as light verbs (e.g. 获得, 发生, etc.)

   Note that for news data (TempEval2 and Wikinews), the news titles and date lines should be ignored.

A.1.5 Event Classification

The fifth pass of annotation is to give each event a label, marking their eventuality types explicitly to help the next stage, event reference time resolution. Following previous work on this topic, we define this stage an eight way classification problem. The eight categories are: Event, Completed Event, Modalized Event, State, Habitual, Ongoing Event, Generic State, Generic Habitual. Classification is based on both the verb itself and its context.

The main distinction that needs to be made clear here is between eventive events and stative events. If an event is emphasizing a process/change of state, it belongs to the Event category. If an event

is describing a property/state of an entity or the world, it is a stative event. Stative events include verbs that are inherently states, for example “有”, “是”, which belong to the State category; and also verbs that are usually used to express an eventive event, but converted to a stative event by emphasizing the result, progressing, possibility, and regularity of the event, which correspondingly map to CompletedEvent, OngoingEvent, ModalizedEvent, and Habitual. In other words, all categories except for Event are stative, describing a property or state of an entity or the world. The key is to make clear which aspect of the event is emphasized.

Please note that in this guidelines, I use the initial-capitalized “Event” to denote the eventuality category “Event”, and all-lower-case “event” to denote all event markables which can be any one of the eight categories.

1. Event

   Predicates that emphasize a change of state with eventive verbs are given the label Event. The following types of events are usually considered as Event.

   - An event that is somebody reporting something is mostly emphasizing the processing of reporting, and hence Event.
     - people + "表示", "显示", "说", "指出", "强调", etc.

2. Completed Event

   Predicates that emphasize the result of changing with eventive verbs are given the label Completed Event. These events describe a state more than a change or a process. The following types of events are usually considered as Completed Event.

   - An event that has happened on / has been done by a group of entities is mostly emphasizing the result of this group of entities having had the experience, hence Completed

Event.

- 香港的长江实业等一批财团纷纷到珠海落户。
- 近几年台湾厂商也在广东设立七千多家加工企业。
- “八五”期间，中国共批准外商投资项目…

- A comparison of states is usually Completed Event.
  - 比…提高..., 比…增长…
  - “八五”期间中国进出口总额达一点零一万亿美元，比“七五”时期增长一倍以上。
- A predicate with 已, 到, 至, 了, 成is more likely to be a Completed Event.
  - 成为, 成了, 形成, 完成, etc.
  - 已渗入到, 上升至, 创下了, etc.
  - 中外经济技术合作与交流已渗入到中国经济生活的各个领域，一个全国范围的大开放格局初步形成。
  - 其中，将于今年七月在全国实施的加工贸易保障金制度成为港澳客商的关注焦点。
- Predicates with the format “有+ nominalized verb”
  - 有上升

3. Modalized Events:

Predicates with modal verbs are Modalized events.

- 投产后，它既可生产与彩电配套的彩色显像管玻壳，…
- 住房制度改革要着眼于建立福利分配货币化的新机制。

Please note that we don't consider future events as modalized events. Future events are labeled as regular events with "Event", "State", "Habitual", etc. labels.

An exception is with the verb "达". When used as "可达", which is ubiquitous in news data, consider it as State instead of ModalizedEvent.

- 全年(timex) 实际利用外资预计可达(state) 二十八亿美元，增长(completed) 百分之十五点六。

4. State

Predicates that are describing the state of some entity or the state of the world with stative verbs are given the label State. The following types of events are usually considered as States.

- Predicates with inherently stative verbs are States.

  Some common stative verbs are:

  - 是, 有, 为, 居, 占, etc.
  - 支持, 接近, 显露, 欢迎, etc.
  - 认为, 估计, 预计, 期望, 以期, 可望, 预示, 有望, etc.
  - 准备, 计划, etc.
  - 达, 达到, etc.

  Some examples:

  - 去年广东省整个加工贸易出口值达四百多亿美元。
  - 目前从该基金中拿出十万美元正在进行中国晖春边境合作区环境评估项目，以期对外资大规模进入这个地区提供环境方面的咨询。

  - “欢迎国际社会同我们一道，共同推进图们江开发事业，促进区域经济发展，造福东北亚人民。”

- Predicates that are a single adjective with the verb dropped are usually States. They usually have the POS tag VA.
  - 速度快、效益好
- Predicates with their verbs dropped and only having the object are mostly States.
  - 目前全省从事加工贸易人数五百多万(state)，仅“三来一补”的企业就达(state)三万多家，从业人员二百多万(state)。
- Predicates that describe the property of an entity or the world that happens every year, every month, every day, ⋯, every second, etc. are States.
  - 年均增长百分之八
- Negated events are mostly describing a state of not doing something, hence State.
  - 跨国公司在山东投资势头不减。
  - 实行保证金台帐制度后，海关对正常开展加工贸易的企业不再征收与进口料件税款等值的风险保证金，只是在银行设立台帐时收取一百元的手续费，因而将减轻企业的实际经济负担。
- An event that is some news, articles, papers, etc. reporting something is mostly emphasizing the statement of the reported content, hence State:
  - news, articles, papers + "表示", "显示", "说", "指出", "强调", etc.
  - 为⋯所+ verb

5. Habitual

Predicates that describe a regularly repeated event are Habituals. The following types of events are common Habitual events.

- In sentences where the events in the main clause are conditioned on the events in a time adverbial clause, both are Habituals.
  - 只是在银行设立台帐时收取一百元的手续费，因而将减轻企业的实际经济负担。
- Predicates that describe an action or behavior that's done repetitively in a given period of time are considered Habituals.
  - 深圳、惠州成为电子工业的出口基地，两市生产的电脑元器件大量销往世界各地。
  - "八五"期间，国民经济更加广泛的参与国际分工与国际交换。

6. Ongoing Events

   Predicates whose verbs are modified by "正在", "日益", "日臻", etc. are usually Ongoing Events.

7. Generic State

   State events with generic subjects are Generic State.

8. Generic Habitual

   Habitual events with generic subjects are Generic Habitual.

A.1.6 Event Reference Time Resolution

The sixth and last pass of annotation is to find the reference time for each event, forming a final parse tree with the children nodes temporally dependent on their parent nodes.

An event's reference time can be either a time or another event. When an event's reference time is a time, i.e. DCT, PAST_REF, PRESENT_REF, FUTURE_REF, or a timex in the text, annotate a link from the time to the event. Most relations between a time and an event are "happened at/around", so no link label is needed for these links. However, for some rare cases that are timex + "以来", "以前", etc., for example "1997年以来", we label the link between the timex ("1997年") and the events happened after ("以来") or before ("以前") the timex as "before" or "after". When an event's reference time is another event in the text, we annotate a link from the reference time event to the current event, and give a link label to this relation. Possible relations between events are: "before", "includes", "overlap", and "after". Here "overlap" is undirected while the other three are directed.

An event is linked to ATEMPORAL if it can't be temporally located or it holds true for the entire timeline. Events are linked to the most specific reference time available (i.e. the lowest node in the parse). For example, if there's a link "DCT →3月18日", then an event happened on that day should be linked to "3月18日" instead of DCT, and another event that happened right after that should be linked to the first event instead of "3月18日".

In this pass of annotation, we start by reading each event in their linear order in the sentences. By default, we assume these events' temporal locations are in the same linear order. Therefore, we start with a parse that looks like a linked list:

e1 →before/overlap →e2 →before/overlap →e3 →before/overlap →⋯
→before/overlap →en

While reading through the events, we look for some events that are "temporal location changers". There are two types of temporal location changers. The first type we call it "temporal location

jumpers". They usually change the linear advance of time by referring to a timex or PRESENT_ref / PAST_ref / FUTURE_ref / atemporal. They are events that jump onto a different temporal location on the timeline. Therefore, a parse with a temporal location jumper would look like this if the "jumping" is caused by referring to a timex:

timex1 →e1 →before/overlap →···→em

timex2 →em+1 →before/overlap →···→en

or this if the "jumping" is caused by referring to PRESENT_ref / PAST_ref / FUTURE_ref / Atemporal:

timex1 →e1 →before/overlap →···→em

PAST_ref →em+1 →before/overlap →···→en

The second type we call it "temporal location advancers". They usually advance the time by mentioning an event that happened a little bit later than a previously mentioned event. If all events happen in the same order with their mentions in the discourse, a parse with temporal location advancer would look like the same with our assumption:

e1 →before/overlap →e2 →before/overlap →e3 →before/overlap →···
→before/overlap →en

When an event advances a temporal location that has several events overlapping on it, the parse with temporal location advancer would look like this:

e1 →overlap →e2 →overlap →···→overlap →em

ek(1<=k<=m) →before →em+1 →before/overlap →···→before/overlap →en

Here, ek is the most related event (usually the last em) to em+1.

In conclusion, events that break the original linear order in the discourse, i.e. temporal location changers, are listed in Table A.3.

Temporal Location Jumpers	Events that refer to a different timex
	Events that refer to Present/Past/Future_Ref/Atemporal
	Events that refer to a previously mentioned event (not immediate previous one)
Temporal Location Advancers	Events that happen after an immediate previously mentioned event

Table A.3: Common events that are temporal location jumpers and advancers.

All events that are NOT temporal location changers are linked to their immediate previous event and labeled "overlap". For example,

- 今天(timex)，由中国贸易促进会广东分会与香港中华总商会联合主办的" 九六税改及进口原料台帐制执行实务研讨会" 在广州举行(event)，向港澳及内地客商介绍(event) 今年国家将实行的外经贸三大政策调整。

  Links: 今天→举行→overlap →介绍

- 去年(timex) 广东省整个加工贸易出口值达(state) 四百多亿美元，约占(state) 全省出口总值的百分之八十。其中一般贸易的一半、来料加工的百分之百和外商投资企业的百分之八十以上属(state) 加工贸易出口。

  Links: 去年→达→overlap →占→overlap →属

- 目前(timex) 全省从事加工贸易人数五百多万(state)，仅“ 三来一补” 的企业就达(state) 三万多家，从业人员二百多万(state) 。

  Links: “目前” →“五百多万” →overlap →“达” →overlap →“二百多万”

- 据介绍，近年(timex) 来广东省对外经贸迅速发展(event)，而包括来料加工、进料加工和外商投资企业从事的加工贸易占(state) 了相当大的份额。

  Links: 近年→发展→overlap →占

A.1.7 Specifications on Some Common Scenarios

1. Q: If an event advances the time after a series of overlapping events, which previously mentioned event do I use as the reference time?

   A: Use the most closely related event (usually the last event). For example,

   - 今天(timex)，由中国贸易促进会广东分会与香港中华总商会联合主办的“ 九六税改及进口原料台帐制执行实务研讨会” 在广州举行(event)，向港澳及内地客商介绍(event) 今年国家将实行的外经贸三大政策调整。将于今年七月(timex) 在全国实施的加工贸易保证金台帐制度成为(event) 港澳客商的关注焦点。

     成为is more closely related to 介绍then 举行, so – Links: 今天→举行→overlap →介绍→before →成为.

2. For scenarios with reporting events, (1) the reporting event’s reference time should be either a timex in the text, PRESENT_ref, or a previously mentioned reporting event that happens either before or overlap with the current reporting event; and (2) the reported content events should take the reporting event as their reference time, and be linked to the reporting event with “before”, “after”, or “overlap”, depending on if the reporting happens before, after, or at

around the same time with the content events. Timex inside the reporting content are ignored because (1) a loose temporal relation is captured here by labeling the before/after/overlap relations between the content events and the reporting event, and (2) the reporting event is the main event that advances the progress of time in the discourse and hence our interests to build into the final parse, while reported content events might be some isolated points on the timeline that are out of the scope of this work.

An example of a reporting event is as follows:

- 广东省外经贸委有关负责人指出(event)，实行加工贸易台帐制度是(state) 为了完善对加工贸易的监管，堵塞管理漏洞，防止国家税收流失，促进加工贸易的健康发展。

The links are: PRESENT_REF →“指出” →overlap →“是”.

3. For scenarios with time adverbial clauses, if both events inside the time adverbial clause and in the main clause are Habituals, meaning when A happens B happens, then link the main clause predicate to the temporal adverbial clause predicate with “overlap”; if the sentence is not describing habitual activities, then link events in the main clause to events in the time adverbial clause with according relation labels. For example,

   - 实行(event) 保证金台帐制度后，海关对正常开展加工贸易的企业不再征收(state) 与进口料件税款等值的风险保证金，只是在银行设立(habitual) 台帐时收取(habitual) 一百元的手续费，因而将减轻(event) 企业的实际经济负担。

   The links are: FUTURE_REF →“不再征收”; “不再征收” →after →“实行” “不再征收” →overlap →“收取” →overlap →“设立”; “收取” →overlap →“减轻”.

4. Note that NOT all timex marked out in the discourse will serve as a reference time. There will be timex that don't have any events linked to them. For example, timex governed by reporting verbs are ignored and are not the reference time for any events. Some other examples are as follows:

   - 当今世界先进技术水平的显像管
   - 硬件配置和软件应用都达到了当今国际同行业先进水平
   - 中国已经确定了未来五年高技术研究重点，并着手制定下世纪的高科技研究计划
   - 比1990年提高了十个百分点
   - 工业产品的制造水平比过去有了很大提高

5. For cases with an aspectual verb followed by a main verb, both are marked out as separate events. We link the aspectual verb to the main verb as an "overlap" relation.

   - "开始", "停止", "保持", etc. →overlap →an event

6. If there are two adjacent timex expressing the same temporal location, link events to the first one.

   - e.g. 广东"八五(timex)"期间（一九九一至一九九五年(timex)）电子工业新兴产业发展速度快(state)、效益好(state)。

   八五→includes →快→overlap →好

7. For cases with "timex + 以来, 以前", etc., for example:

   - timex 以来，state/event: timex →before →state/event

- timex 以前，state/event: timex →after →state/event

8. For scenarios with adverbial “预计”, “估计”, etc., don’t consider them as events. For example:

   - 全年(timex) 实际利用外资预计可达(state) 二十八亿美元，增长(completed) 百分之十五点六。

     全年→达→overlap →增长

   - 全市今年(timex) 工业总产值预计达(state) 一千二百五十五亿元，增长(completed) 百分之十七点五。

     今年→达→overlap →增长

9. For cases with the pattern “event1 …使, 导致, 以, 去做, etc. …event2”, consider these two events closely related, and parse them as:

   - event1 →before/overlap →event2

10. For quotes that are not preceded or governed by a reporting verb, annotated them as regular events outside a quote. For example,

    - “ 中国国家气象局购买(event) 美国克雷公司的大型计算机，克雷公司只卖给(event) 我们两台处理器。如今(timex) ，我们自己开发的‘曙光１０００’计算机每秒最大计算速度已达(state) 二十五亿次，超过(state) 克雷公司卖给我们的计算机速度。” 这是国务委员兼国家科委主任宋健今天在“ 八六三计划” 工作会议上讲的一段话。

    Here, events “购买”, “卖给”, “达”, “超过” are in a quote without a reporting verb, so simply link them as if they were not in a quote:

PAST_REF →购买→overlap →卖给; 如今→达→overlap →超过;

11. Most events' reference times should be before the events in the discourse. For example,

    - 重庆科技人才优势在中国西部比较突出，现有各类科研机构三百四十个，大专院校二十三所，各类科技人才三十五万多人。

    link "PRESENT_ref →突出; 现→有" instead of "现→突出→overlap →有".

    Predicates that express the pass of a period of time are not considered as events. For example,

    - １９３９年９月１日，希特勒军队进攻波兰，开始了给欧洲和世界人民带来巨大痛苦的第二次世界大战，迄今正好５５年。
    - 这次战争中包括德国在内的欧洲国家共死亡４００万人。
    - 其中，苏联为战胜德国侵略者作出了巨大牺牲，伤亡近２８００万人。
    - 半个世纪转瞬逝去(not event)，欧洲经历了热战、冷战、动荡、冲突和剧变。

    here, 逝去is expressing the passing of half a century and is not considered as an event.

12. Consider modalized events with "要", "要求", etc. as events on their happening temporal locations, instead of states of reporting temporal locations. For example:

    - 宋健重申，在淮河流域范围内，禁止新建小造纸、小化工、小制革等污染严重的项目。对所有向淮河流域河流排污的企业，要进行限期治理，

    link "重申→overlap →禁止→before →治理" instead of "重申→overlap →禁止→overlap →治理".

13. Special treatment for narrative discourses

- Time expressions
  - past_ref →从前
  - 从前→includes →有一天
  - 有一天→before →第二天→before →第三天
- Posit events in the past by default.
  - past_ref →first event in the document
- Time advancing events/states vs. non-time-advancing states
  - Time advancing events/states form the main timeline, and they are linked to each other by “before/overlap”.
  - Non-time-advancing states form the branch timelines, and they are linked to the main timeline by “overlap/after”.

  男人觉得这很好，他说：“.....”
- Imperatives in quotes – not events

Bibliography

[Abdulsalam et al., 2016] Abdulsalam, A. A., Velupillai, S., and Meystre, S. (2016). Utahbmi at semeval-2016 task 12: extracting temporal information from clinical text. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1256–1262.

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

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Barros et al., 2016] Barros, M., Lamúrias, A., Figueiró, G., Antunes, M., Teixeira, J., Pinheiro, A., and Couto, F. M. (2016). Ulisboa at semeval-2016 task 12: Extraction of temporal expressions, clinical events and relations using ibent. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1263–1267.

[Bethard, 2013] Bethard, S. (2013). Cleartk-timeml: A minimalist approach to tempeval 2013. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 10–14.

[Bethard et al., 2015a] Bethard, S., Derczynski, L., Savova, G., Pustejovsky, J., and Verhagen, M. (2015a). Semeval-2015 task 6: Clinical tempeval. In *SemEval@ NAACL-HLT*, pages 806–814.

[Bethard et al., 2015b] Bethard, S., Derczynski, L., Savova, G., Pustejovsky, J., and Verhagen, M. (2015b). Semeval-2015 task 6: Clinical tempeval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814.

[Bethard et al., 2012] Bethard, S., Kolomiyets, O., and Moens, M.-F. (2012). Annotating narrative timelines as temporal dependency structures. In *Proceedings of the International Conference on Linguistic Resources and Evaluation, Istanbul, Turkey, May. ELRA*.

[Bethard and Martin, 2007] Bethard, S. and Martin, J. H. (2007). Cu-tmp: Temporal relation classification using syntactic and semantic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 129–132. Association for Computational Linguistics.

[Bethard et al., 2007] Bethard, S., Martin, J. H., and Klingenstein, S. (2007). Timelines from text: Identification of syntactic temporal relations. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 11–18. IEEE.

[Bethard et al., 2016a] Bethard, S., Savova, G., Chen, W.-T., Derczynski, L., Pustejovsky, J., and Verhagen, M. (2016a). Semeval-2016 task 12: Clinical tempeval. *Proceedings of SemEval*, pages 1052–1062.

[Bethard et al., 2016b] Bethard, S., Savova, G., Chen, W.-T., Derczynski, L., Pustejovsky, J., and Verhagen, M. (2016b). Semeval-2016 task 12: Clinical tempeval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062.

[Bethard et al., 2017] Bethard, S., Savova, G., Palmer, M., and Pustejovsky, J. (2017). Semeval-2017 task 12: Clinical tempeval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572, Vancouver, Canada. Association for Computational Linguistics.

[Blitzer et al., 2006] Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics.

[Bohnemeyer, 2009] Bohnemeyer, J. (2009). Temporal anaphora in a tenseless language. *The expression of time in language*, pages 83–128.

[Brewer and Lichtenstein, 1982] Brewer, W. F. and Lichtenstein, E. H. (1982). Stories are to entertain: A structural-affect theory of stories. *Journal of pragmatics*, 6(5-6):473–486.

[Caselli and Morante, 2016] Caselli, T. and Morante, R. (2016). Vuacltl at semeval 2016 task 12: A crf pipeline to clinical tempeval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1241–1247.

[Cassidy et al., 2014] Cassidy, T., McDowell, B., Chambers, N., and Bethard, S. (2014). An annotation framework for dense event ordering. In *The 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, June. Association for Computational Linguistics*.

[Chambers, 2013] Chambers, N. (2013). Navytime: Event and time ordering from raw text. Technical report, DTIC Document.

[Chambers et al., 2014] Chambers, N., Cassidy, T., McDowell, B., and Bethard, S. (2014). Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

[Chambers and Jurafsky, 2008a] Chambers, N. and Jurafsky, D. (2008a). Jointly combining implicit constraints improves temporal ordering. In *EMNLP-2008*.

[Chambers and Jurafsky, 2008b] Chambers, N. and Jurafsky, D. (2008b). Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 698–706. Association for Computational Linguistics.

[Chambers et al., 2007] Chambers, N., Wang, S., and Jurafsky, D. (2007). Classifying temporal relations between events. In *ACL-2007*.

[Chang and Manning, 2013] Chang, A. and Manning, C. D. (2013). Sutime: Evaluation in tempeval-3. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 78–82.

[Cheng et al., 2007] Cheng, Y., Asahara, M., and Matsumoto, Y. (2007). Naist. japan: Temporal relation identification using dependency parsed tree. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 245–248. Association for Computational Linguistics.

[Chikka, 2016] Chikka, V. R. (2016). Cde-iiith at semeval-2016 task 12: Extraction of temporal information from clinical documents using machine learning techniques. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1237–1240.

[Chu and Liu, 1965] Chu, Y.-J. and Liu, T. H. (1965). On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

[Cohan et al., 2016] Cohan, A., Meurer, K., and Goharian, N. (2016). Guir at semeval-2016 task 12: Temporal information processing for clinical narratives. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1248–1255.

[Covington, 2001] Covington, M. A. (2001). A fundamental algorithm for dependency parsing. In *Proceedings of the 39th annual ACM southeast conference*, pages 95–102. Citeseer.

[Crammer et al., 2006] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585.

[Crammer and Singer, 2003] Crammer, K. and Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3(Jan):951–991.

[Cui et al., 2017] Cui, Y., Liu, T., Chen, Z., Ma, W., Wang, S., and Hu, G. (2017). Dataset for the first evaluation on chinese machine reading comprehension. *arXiv preprint arXiv:1709.08299*.

[Cui et al., 2016] Cui, Y., Liu, T., Chen, Z., Wang, S., and Hu, G. (2016). Consensus attention-based neural networks for chinese reading comprehension. *arXiv preprint arXiv:1607.02250*.

[Cui et al., 2018] Cui, Y., Liu, T., Xiao, L., Chen, Z., Ma, W., Che, W., Wang, S., and Hu, G. (2018). A span-extraction dataset for chinese machine reading comprehension. *arXiv preprint arXiv:1810.07366*.

[Daumé III, 2009] Daumé III, H. (2009). Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

[Derczynski and Gaizauskas, 2010] Derczynski, L. and Gaizauskas, R. (2010). Usfd2: Annotating temporal expresions and tlinks for tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 337–340. Association for Computational Linguistics.

[Dligach et al., 2017] Dligach, D., Miller, T., Lin, C., Bethard, S., and Savova, G. (2017). Neural temporal relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 746–751.

[Do et al., 2012] Do, Q. X., Lu, W., and Roth, D. (2012). Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687. Association for Computational Linguistics.

[Edmonds, 1967] Edmonds, J. (1967). Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71:233–240.

[Ferro et al., 2001] Ferro, L., Mani, I., Sundheim, B., and Wilson, G. (2001). Tides temporal annotation guidelines version 1.0. 2. *The MITRE Corporation, McLean-VG-USA*.

[Filannino et al., 2013] Filannino, M., Brown, G., and Nenadic, G. (2013). Mantime: Temporal expression identification and normalization in the tempeval-3 challenge. *arXiv preprint arXiv:1304.7942*.

[Finin et al., 2010] Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., and Dredze, M. (2010). Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88. Association for Computational Linguistics.

[Fries, 2016] Fries, J. A. (2016). Brundlefly at semeval-2016 task 12: Recurrent neural networks vs. joint inference for clinical temporal information extraction. *arXiv preprint arXiv:1606.01433*.

[Georgiadis, 2003] Georgiadis, L. (2003). Arborescence optimization problems solvable by edmonds' algorithm. *Theoretical Computer Science*, 301(1-3):427–437.

[Grouin and Moriceau, 2016] Grouin, C. and Moriceau, V. (2016). Limsi at semeval-2016 task 12: machine-learning and temporal information to identify clinical events and time expressions. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1225–1230.

[Grover et al., 2010] Grover, C., Tobin, R., Alex, B., and Byrne, K. (2010). Edinburgh-ltg: Tempeval-2 system description. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 333–336.

[Ha et al., 2010] Ha, E. Y., Baikadi, A., Licata, C., and Lester, J. C. (2010). Ncsu: modeling temporal relations with markov logic and lexical ontology. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 341–344. Association for Computational Linguistics.

[Hagège and Tannier, 2007] Hagège, C. and Tannier, X. (2007). Xrce-t: Xip temporal module for tempeval campaign. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 492–495.

[Hansart et al., 2016] Hansart, C., De Meyere, D., Watrin, P., Bittar, A., and Fairon, C. (2016). Cental at semeval-2016 task 12: A linguistically fed crf model for medical and temporal information extraction. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1286–1291.

[Hayes and Krippendorff, 2007] Hayes, A. F. and Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89.

[He et al., 2018] He, W., Liu, K., Liu, J., Lyu, Y., Zhao, S., Xiao, X., Liu, Y., Wang, Y., Wu, H., She, Q., et al. (2018). Dureader: a chinese machine reading comprehension dataset from real-world applications. *ACL 2018*, page 37.

[Hepple et al., 2007] Hepple, M., Setzer, A., and Gaizauskas, R. (2007). Usfd: preliminary exploration of features and classifiers for the tempeval-2007 tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 438–441. Association for Computational Linguistics.

[Hinrichs, 1986] Hinrichs, E. (1986). Temporal anaphora in discourses of english. *Linguistics and philosophy*, 9(1):63–82.

[Hinrichs, 1981] Hinrichs, E. W. (1981). Temporale anaphora in englischen. *StaatsExamen Thesis*.

[Hitzeman et al., 1995] Hitzeman, J., Moens, M., and Grover, C. (1995). Algorithms for analysing the temporal structure of discourse. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, pages 253–260. Morgan Kaufmann Publishers Inc.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Huang et al., 2017] Huang, P.-Y., Huang, H.-H., Wang, Y.-W., Huang, C., and Chen, H.-H. (2017). Ntu-1 at semeval-2017 task 12: detection and classification of temporal events in clinical data with domain adaptation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1010–1013.

[Johnson-Laird, 1980] Johnson-Laird, P. N. (1980). Mental models in cognitive science. *Cognitive science*, 4(1):71–115.

[Jung and Stent, 2013] Jung, H. and Stent, A. (2013). Att1: Temporal annotation using big windows and rich syntactic and semantic features. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 20–24.

[Kamp, 1979] Kamp, H. (1979). Events, instants and temporal reference. In *Semantics from different points of view*, pages 376–418. Springer.

[Kiperwasser and Goldberg, 2016] Kiperwasser, E. and Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional lstm feature representations. *arXiv preprint arXiv:1603.04351*.

[Kitaev and Klein, 2018] Kitaev, N. and Klein, D. (2018). Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics.

[Kolomiyets et al., 2012] Kolomiyets, O., Bethard, S., and Moens, M.-F. (2012). Extracting narrative timelines as temporal dependency structures. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 88–97. Association for Computational Linguistics.

[Kolomiyets and Moens, 2010] Kolomiyets, O. and Moens, M.-F. (2010). Kul: recognition and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 325–328. Association for Computational Linguistics.

[Kolomiyets and Moens, 2013] Kolomiyets, O. and Moens, M.-F. (2013). Kul: data-driven approach to temporal parsing of newswire articles. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 83–87.

[Kolya et al., 2010] Kolya, A. K., Ekbal, A., and Bandyopadhyay, S. (2010). Ju_cse_temp: A first step towards evaluating events, time expressions and temporal relations. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 345–350. Association for Computational Linguistics.

[Kolya et al., 2013] Kolya, A. K., Kundu, A., Gupta, R., Ekbal, A., and Bandyopadhyay, S. (2013). Ju_cse: A crf based approach to annotation of temporal expression, event and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 64–72.

[Krippendorff, 2004] Krippendorff, K. (2004). *Content analysis: An introduction to its methodology*. Sage Publications, Inc.

[Lamurias et al., 2017] Lamurias, A., Sousa, D., Pereira, S., Clarke, L., and Couto, F. M. (2017). Ulisboa at semeval-2017 task 12: Extraction and classification of temporal expressions and events. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1019–1023.

[Laokulrat et al., 2013] Laokulrat, N., Miwa, M., Tsuruoka, Y., and Chikayama, T. (2013). Uttime: Temporal relation classification using deep syntactic features. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 88–92.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lee et al., 2016] Lee, H.-J., Xu, H., Wang, J., Zhang, Y., Moon, S., Xu, J., and Wu, Y. (2016). Uthealth at semeval-2016 task 12: an end-to-end system for temporal information extraction from clinical notes. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1292–1297.

[Lee et al., 2017] Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

[Leeuwenberg and Moens, 2016] Leeuwenberg, A. and Moens, M.-F. (2016). Kuleuven-liir at semeval 2016 task 12: Detecting narrative containment in clinical records. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1280–1285.

[Leeuwenberg and Moens, 2017] Leeuwenberg, T. and Moens, M.-F. (2017). Structured learning for temporal relation extraction from clinical records. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1150–1158.

[Li et al., 2014] Li, H., Strötgen, J., Zell, J., and Gertz, M. (2014). Chinese temporal tagging with heideltime. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 133–137. Association for Computational Linguistics.

[Li and Huang, 2016] Li, P. and Huang, H. (2016). Uta dlnlp at semeval-2016 task 12: deep learning based natural language processing system for clinical information identification from clinical notes and pathology reports. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1268–1273.

[Li et al., 2016] Li, P., Li, W., He, Z., Wang, X., Cao, Y., Zhou, J., and Xu, W. (2016). Dataset and neural recurrent sequence labeling model for open-domain factoid question answering. *arXiv preprint arXiv:1607.06275*.

[Lin et al., 2015] Lin, C., Dligach, D., Miller, T. A., Bethard, S., and Savova, G. K. (2015). Multilayered temporal modeling for the clinical domain. *Journal of the American Medical Informatics Association*, 23(2):387–395.

[Lin et al., 2016] Lin, C., Miller, T., Dligach, D., Bethard, S., and Savova, G. (2016). Improving temporal relation extraction with training instance augmentation. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 108–113.

[Llorens et al., 2010] Llorens, H., Saquete, E., and Navarro, B. (2010). Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291. Association for Computational Linguistics.

[Llorens et al., 2013] Llorens, H., Saquete, E., and Navarro-Colorado, B. (2013). Applying semantic knowledge to the automatic processing of temporal expressions and events in natural language. *Information Processing & Management*, 49(1):179–197.

[Lloret et al., 2013] Lloret, E., Plaza, L., and Aker, A. (2013). Analyzing the capabilities of crowdsourcing services for text summarization. *Language resources and evaluation*, 47(2):337–369.

[Long et al., 2017] Long, Y., Li, Z., Wang, X., and Li, C. (2017). Xjnlp at semeval-2017 task 12: Clinical temporal information ex-traction with a hybrid model. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1014–1018.

[MacAvaney et al., 2017] MacAvaney, S., Cohan, A., and Goharian, N. (2017). Guir at semeval-2017 task 12: A framework for cross-domain clinical temporal information extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1024–1029.

[Manfredi et al., 2014] Manfredi, G., Strötgen, J., Zell, J., and Gertz, M. (2014). eideltime at eventi: Tuning italian resources and addressing timeml's empty tags. In *Proceedings of the Forth International Workshop EVALITA*, pages 39–43.

[Mani and Pustejovsky, 2004] Mani, I. and Pustejovsky, J. (2004). Temporal discourse models for narrative structure. In *Proceedings of the 2004 ACL Workshop on Discourse Annotation*, pages 57–64. Association for Computational Linguistics.

[Manning et al., 2014] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

[McCawley, 1971] McCawley, J. D. (1971). Tense and time reference in english. *Studies in Linguistic Semantics*.

[McClosky et al., 2010] McClosky, D., Charniak, E., and Johnson, M. (2010). Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 28–36. Association for Computational Linguistics.

[Miller et al., 2013] Miller, T. A., Bethard, S., Dligach, D., Pradhan, S., Lin, C., and Savova, G. K. (2013). Discovering narrative containers in clinical text. *ACL 2013*, page 18.

[Min et al., 2007] Min, C., Srikanth, M., and Fowler, A. (2007). Lcc-te: a hybrid approach to temporal relation identification in news text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 219–222. Association for Computational Linguistics.

[Mostafazadeh et al., 2016] Mostafazadeh, N., Grealish, A., Chambers, N., Allen, J., and Vanderwende, L. (2016). Caters: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the The 4th Workshop on EVENTS: Definition, Detection, Coreference, and Representation, San Diego, California, June. Association for Computational Linguistics*.

[Neubig et al., 2017] Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulos, A., Ballesteros, M., Chiang, D., Clothiaux, D., Cohn, T., Duh, K., Faruqui, M., Gan, C., Garrette, D., Ji, Y., Kong, L., Kuncoro, A., Kumar, G., Malaviya, C., Michel, P., Oda, Y., Richardson, M., Saphra, N., Swayamdipta, S., and Yin, P. (2017). Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

[Ng and Kan, 2012] Ng, J.-P. and Kan, M.-Y. (2012). Improved temporal relation classification using dependency parses and selective crowdsourced annotations. *Proceedings of COLING 2012*, pages 2109–2124.

[Ning et al., 2018a] Ning, Q., Wu, H., Peng, H., and Roth, D. (2018a). Improving temporal relation extraction with a globally acquired statistical resource. *arXiv preprint arXiv:1804.06020*.

[Ning et al., 2018b] Ning, Q., Wu, H., and Roth, D. (2018b). A multi-axis annotation scheme for event temporal relations. *arXiv preprint arXiv:1804.07828*.

[O'Gorman et al., 2016] O'Gorman, T., Wright-Bettner, K., and Palmer, M. (2016). Richer event description: Integrating event coreference with temporal, causal and bridging annotation. *Computing News Storylines*, page 47.

[Parker et al., 2011] Parker, R., Graff, D., Kong, J., Chen, K., and Maeda, K. (2011). English gigaword fifth edition. *LDC Catalog Ref. LDC2011T07*.

[Partee, 1973] Partee, B. H. (1973). Some structural analogies between tenses and pronouns in english. *The Journal of Philosophy*, 70(18):601–609.

[Partee, 1984] Partee, B. H. (1984). Nominal and temporal anaphora. *Linguistics and philosophy*, 7(3):243–286.

[Partes, 1984] Partes, B. H. (1984). Nominal and temporal anaphora. *Linguistics and philosophy*, 7(3):243–286.

[Puşcaşu, 2007] Puşcaşu, G. (2007). Wvali: Temporal relation identification by syntactico-semantic analysis. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 484–487. Association for Computational Linguistics.

[Pustejovsky et al., 2003a] Pustejovsky, J., Castano, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., Katz, G., and Radev, D. R. (2003a). Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

[Pustejovsky et al., 2003b] Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., et al. (2003b). The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

[Pustejovsky and Stubbs, 2011] Pustejovsky, J. and Stubbs, A. (2011). Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160. Association for Computational Linguistics.

[Rajpurkar et al., 2018] Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

[Reichenbach, 1947] Reichenbach, H. (1947). *Elements of Symbolic Logic*. The MacMillan Company, New York.

[Saquete, 2010] Saquete, E. (2010). Id 392: Terseo+ t2t3 transducer: a systems for recognizing and normalizing timex3. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 317–320. Association for Computational Linguistics.

[Sarath et al., 2016] Sarath, P., Manikandan, R., and Niwa, Y. (2016). Hitachi at semeval-2016 task 12: A hybrid approach for temporal information extraction from clinical notes. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1231–1236.

[Sarath et al., 2017] Sarath, P., Manikandan, R., and Niwa, Y. (2017). Hitachi at semeval-2017 task 12: System for temporal information extraction from clinical notes. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1005–1009.

[Setzer, 2002] Setzer, A. (2002). *Temporal information in newswire articles: an annotation scheme and corpus study.* PhD thesis, University of Sheffield.

[Shao et al., 2018] Shao, C. C., Liu, T., Lai, Y., Tseng, Y., and Tsai, S. (2018). Drcd: a chinese machine reading comprehension dataset. *arXiv preprint arXiv:1806.00920*.

[Smith, 2003] Smith, C. S. (2003). *Modes of discourse: The local structure of texts*, volume 103. Cambridge University Press.

[Snow et al., 2008] Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics.

[Steedman, 1982] Steedman, M. J. (1982). Reference to past time. *Speech, place and action*, pages 125–157.

[Strötgen et al., 2014] Strötgen, J., Armiti, A., Van Canh, T., Zell, J., and Gertz, M. (2014). Time for more languages: Temporal tagging of arabic, italian, spanish, and vietnamese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 13(1):1–21.

[Strötgen and Gertz, 2010] Strötgen, J. and Gertz, M. (2010). Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324. Association for Computational Linguistics.

[Strötgen and Gertz, 2015] Strötgen, J. and Gertz, M. (2015). A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 541–547, Lisbon, Portugal. Association for Computational Linguistics.

[Strötgen et al., 2013] Strötgen, J., Zell, J., and Gertz, M. (2013). Heideltime: Tuning english and developing spanish resources for tempeval-3. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 15–19.

[Styler IV et al., 2014a] Styler IV, W. F., Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P. C., Erickson, B., Miller, T., Lin, C., Savova, G., et al. (2014a). Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

[Styler IV et al., 2014b] Styler IV, W. F., Savova, G., Palmer, M., Pustejovsky, J., O'Gorman, T., and de Groen, P. C. (2014b). Thyme annotation guidelines.

[Sun et al., 2013] Sun, W., Rumshisky, A., and Uzuner, O. (2013). Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

[Tissot et al., 2015] Tissot, H., Gorrell, G., Roberts, A., Derczynski, L., and Del Fabro, M. D. (2015). Ufprsheffield: Contrasting rule-based and support vector machine approaches to time expression identification in clinical tempeval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 835–839.

[Tourille et al., 2016] Tourille, J., Ferret, O., Névéol, A., and Tannier, X. (2016). Limsi-cot at semeval-2016 task 12: Temporal relation identification using a pipeline of classifiers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1136–1142.

[Tourille et al., 2017] Tourille, J., Ferret, O., Tannier, X., and Névéol, A. (2017). Limsi-cot at semeval-2017 task 12: Neural architecture for temporal information extraction from clinical narratives. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 597–602.

[UzZaman and Allen, 2010] UzZaman, N. and Allen, J. F. (2010). Trips and trios system for tempeval-2: Extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 276–283. Association for Computational Linguistics.

[UzZaman et al., 2012] UzZaman, N., Llorens, H., Allen, J., Derczynski, L., Verhagen, M., and Pustejovsky, J. (2012). Tempeval-3: Evaluating events, time expressions, and temporal relations. *arXiv preprint arXiv:1206.5333*.

[Velupillai et al., 2015] Velupillai, S., Mowery, D. L., Abdelrahman, S., Christensen, L., and Chapman, W. (2015). Blulab: Temporal information extraction for the 2015 clinical tempeval challenge. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 815–819.

[Verhagen, 2004] Verhagen, M. (2004). Times between the lines. *Brandeis University, Massachusetts*.

[Verhagen et al., 2007a] Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., and Pustejovsky, J. (2007a). Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80. Association for Computational Linguistics.

[Verhagen et al., 2007b] Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., and Pustejovsky, J. (2007b). Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th international workshop on semantic evaluations*, pages 75–80. Association for Computational Linguistics.

[Verhagen et al., 2009] Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Moszkowicz, J., and Pustejovsky, J. (2009). The TempEval Challeage: Identifying Temporal Relations in Text. *Lang Resources & Evaluation*, 43:161–179.

[Verhagen et al., 2010a] Verhagen, M., Sauri, R., Caselli, T., and Pustejovsky, J. (2010a). Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 57–62. Association for Computational Linguistics.

[Verhagen et al., 2010b] Verhagen, M., Sauri, R., Caselli, T., and Pustejovsky, J. (2010b). Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 57–62. Association for Computational Linguistics.

[Vicente-Díez et al., 2010] Vicente-Díez, M. T., Schneider, J. M., and Martínez, P. (2010). Uc3m system: Determining the extent, type and value of time expressions in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 329–332. Association for Computational Linguistics.

[Wang and Xue, 2014] Wang, Z. and Xue, N. (2014). Joint pos tagging and transition-based constituent parsing in chinese with non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 733–742.

[Webber, 1987] Webber, B. L. (1987). The interpretation of tense in discourse. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 147–154. Association for Computational Linguistics.

[Webber, 1988] Webber, B. L. (1988). Tense as discourse anaphor. *Computational Linguistics*, 14(2):61–73.

[Wu et al., 2014] Wu, S., Miller, T., Masanz, J., Coarr, M., Halgrim, S., Carrell, D., and Clark, C. (2014). Negation' s not solved: generalizability versus optimizability in clinical natural language processing. *PloS one*, 9(11):e112774.

[Wuyun, 2016] Wuyun, S. (2016). The influence of tense interpretation on discourse coherence - a comparison between mandarin narrative and report discourse. *Lingua*, 179:38 – 56.

[Xue et al., 2010] Xue, N., Jiang, Z., Zhong, X., Palmer, M., Xia, F., Chiou, F.-D., and Chang, M. (2010). Chinese treebank 7.0. *Linguistic Data Consortium, Philadelphia*.

[Xue et al., 2016] Xue, N., Ng, H. T., Pradhan, S., Rutherford, A., Webber, B., Wang, C., and Wang, H. (2016). Conll 2016 shared task on multilingual shallow discourse parsing. *Proceedings of the CoNLL-16 shared task*, pages 1–19.

[Xue et al., 2005] Xue, N., Xia, F., Chiou, F.-D., and Palmer, M. (2005). The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207–238.

[Yao et al., 2019] Yao, J., Feng, M., Feng, H., Wang, Z., Zhang, Y., and Xue, N. (2019). Smart: A stratified machine reading test. *Proceedings of NLPCC 2019*.

[Yoshikawa et al., 2009] Yoshikawa, K., Riedel, S., Asahara, M., and Matsumoto, Y. (2009). Jointly identifying temporal relations with markov logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 405–413. Association for Computational Linguistics.

[Zaidan and Callison-Burch, 2011] Zaidan, O. F. and Callison-Burch, C. (2011). Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1220–1229. Association for Computational Linguistics.

[Zavarella and Tanev, 2013] Zavarella, V. and Tanev, H. (2013). Fss-timex for tempeval-3: Extracting temporal information from text. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 58–63.

[Zhang and Xue, 2014] Zhang, Y. and Xue, N. (2014). Automatic inference of the tense of chinese events using implicit linguistic information. In *EMNLP*, pages 1902–1911. Citeseer.

[Zhang and Xue, 2018a] Zhang, Y. and Xue, N. (2018a). Neural ranking models for temporal dependency structure parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3339–3349.

[Zhang and Xue, 2018b] Zhang, Y. and Xue, N. (2018b). Structured interpretation of temporal relations. In *Proceedings of the International Conference on Linguistic Resources and Evaluation, Miyazaki, Japan, May. ELRA*.

[Zhang and Xue, 2018c] Zhang, Y. and Xue, N. (2018c). Structured interpretation of temporal relations. In *Proceedings of 11th edition of the Language Resources and Evaluation Conference (LREC-2018)*, Miyazaki, Japan.

[Zheng et al., 2019] Zheng, C., Huang, M., and Sun, A. (2019). Chid: A large-scale chinese idiom dataset for cloze test. *arXiv preprint arXiv:1906.01265*.