

Structured Interpretation of Temporal Relations

Yuchen Zhang and Nianwen Xue

Brandeis University
415 South Street, Waltham, MA
yuchenz@brandeis.edu, xuen@brandeis.edu

Abstract

Temporal relations between events and time expressions in a document are often modeled in an unstructured manner where relations between individual pairs of time expressions and events are considered in isolation. This often results in inconsistent and incomplete annotation and computational modeling. We propose a novel annotation approach where events and time expressions in a document form a dependency tree in which each dependency relation corresponds to an instance of temporal anaphora where the antecedent is the parent and the anaphor is the child. We annotate a corpus of 235 documents using this approach in the two genres of news and narratives, with 48 documents doubly annotated. We report a stable and high inter-annotator agreement on the doubly annotated subset, validating our approach, and perform a quantitative comparison between the two genres of the entire corpus. We make this corpus publicly available.

Keywords: Temporal Relation, Dependency Structure, Data Annotation

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., 2014; 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., 2015; Bethard et al., 2016; 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 expres-

sions, 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, 2008; 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:

1. He arrived on Thursday. He got here at 8:00am.

2. He arrived at school, walked to his classroom, and then the class began.

In (1), 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 (2), 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, new 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 detail in Section 4.2.. 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 paper are:

- We propose a novel and comprehensive temporal dependency structure to capture temporal 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 (§3.2.1.).
- 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.

In the next few sections, we will briefly discuss related work (§2.), describe our annotation scheme (§3.), and present our annotation experiments (§4.). We summarize our work in §5.

2. Related Work

Using a dependency structure to represent temporal relations in a document has been proposed before (Kolomiyets et al., 2012), but our work is more comprehensive and linguistically grounded in the following ways. First, their dependency structure is based on events, to the exclusion of time expressions. Time expressions are a strong source of temporal location information for events and excluding them will result in incomplete temporal structures. We cover both events and time expressions to form a complete temporal structure for a text. Second, they exclude stative events such as modalized events, while we provide a more complete temporal structure that include stative events. Third, although they link events in a text to form a dependency structure, they do not explicitly spell out the linguistic basis for the temporal dependencies and annotators are only instructed to identify the most plausible parent for each event. In contrast, we explicitly specify how antecedents of events or time expressions are determined based on a long line of theoretical and computational linguistic research (Reichenbach, 1947; Partee, 1973; Partes, 1984; Hinrichs, 1986; Webber, 1988; Bohnemeyer, 2009; Wuyun, 2016) and these specifications are given to annotators as guidelines when they annotated the data. And lastly, their annotation work is only performed on children’s stories (narrative data), while our annotated corpus covers both news and narrative genres. Annotating two different genres is crucial for us to show that the temporal structure for the two genres are very different, an observation that has implication for automatic parsing strategies.

3. 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. Detailed descriptions for each set are in the following subsections, followed by some examples.

3.1. 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:

3.1.1. 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,

¹https://github.com/yuchenz/structured_temporal_relations.corpus

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 (3) 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 (4) can’t serve as a reference time.

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

4. 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 .

3.1.2. 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 .

3.1.3. 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 11, see §3.2.1. 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 12).

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 5), or a general time expression referring to the past (Example 6). Both cases take “Past_Ref” as their parent.

5. It was a snowy night. [Past_Ref]
6. 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 (example 10), and for vague time expressions, their antecedent is “Present_Ref” (Example 7). See §3.2.1. for more details on time expression classification.

7. China annual economic output results have grown increasingly smooth in recent years. [Present_Ref]
8. Economists who try to estimate actual growth tend to come up with lower numbers. [Present_Ref]
9. China will remain a trade partner as important to Japan as the United States in the future. [Future_Ref]
10. The economy expanded 6.9 percent last year. [DCT]
11. A trend of gradual growth began in 2011. [ROOT]
12. The earth goes around the sun. [Atemporal]

3.2. 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 .

3.2.1. 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 (13), using a pair-wise annotation approach, three temporal relations will be extracted:

13. The economy expanded 6.6 percent in 2003_{t1}, reaching its peak 7.1 percent in March_{t2}. The growth rate doubled in the next year_{t3}.

(2003, includes, March)
(2003, before, next year)
(March, before, next year)

We argue 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 automatically computed. We argue that this notion of reference time for time expressions is intuitive and easy to define. Annotating temporal dependency between time expressions 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 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 (1) 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”).

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.

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

Taxonomy				Examples	Possible Reference Times
Time Expressions	Locatable Time Expressions	Concrete	Absolute	May 2015	ROOT
			Relative	today, two days later	DCT, another Concrete
	Vague		nowadays	Present/Past/Future_Ref	
	Unlocatable Time Expressions		every month	-	

Table 1: Taxonomy of time expressions in our annotation scheme, with examples and possible reference times.

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 (13). 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 1.

3.2.2. 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 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). They 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., 2014), etc. A detailed comparison of these sets can be found in Mostafazadeh et al. (2016). 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 2.

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

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

Table 2: Our temporal relation set for events with mappings to TimeML’s set.

types of antecedents they take. An eventive event usually takes either a time expression or another eventive event 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 (1) 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 (2) 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 (4) 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.

Readers are referred to our more detailed guidelines² on time expression and event recognition, classification, and reference time annotation, which details basic principles for specific cases and discusses extra rules for special scenarios.

3.3. Full Temporal Structure Examples

We present a full example temporal dependency structure for a short news report paragraph (14), as illustrated in Figure 1, and another one for a narrative passage (15), as illustrated in Figure 2. Subscript e denotes eventive events, t denotes time expressions, and s denotes stative events. Un-labeled edges are “depend-on” relations.

²https://github.com/yuchenz/structured_temporal_relations_corpus

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.

14. 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}.³

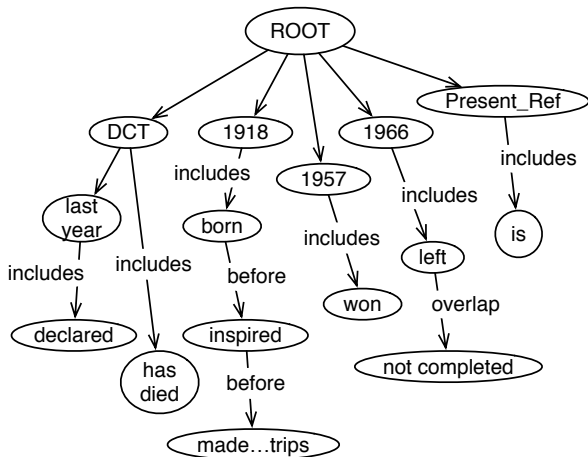


Figure 1: An example full temporal dependency structure for news paragraph (14).

15. 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.⁴

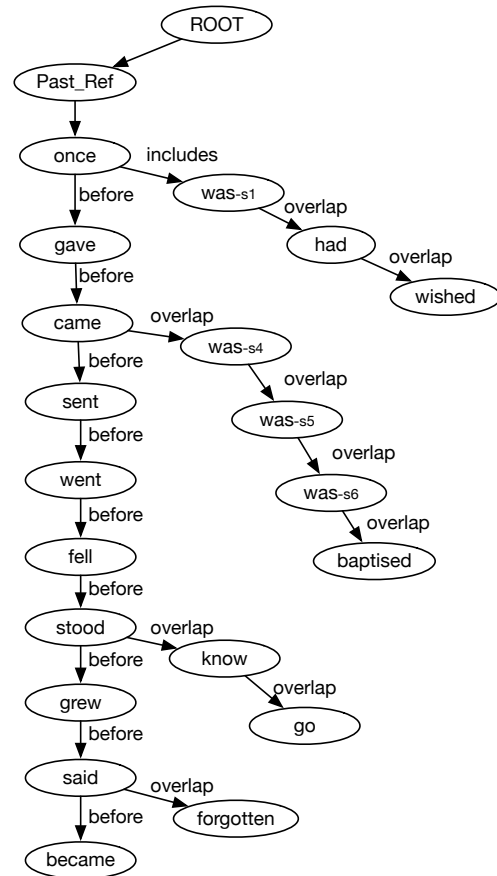


Figure 2: An example full temporal dependency structure for narrative paragraph (15).

3.4. 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.

4. Annotation Analysis

4.1. Corpus

A corpus of 115 news articles, sampled from Chinese TempEval2 data (Verhagen et al., 2010a) and Wikinews data,⁵ and 120 story articles, sampled from Chinese Grimm fairy tales,⁶ are compiled and annotated. 20% of the documents are double annotated by native Chinese speakers. Table 4 presents the detailed statistics. High and stable inter-annotator agreements are reported in Table 5.

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

³From a news report on *The Telegraph*

⁴From Grimm's fairy tale *The Seven Ravens*

⁵zh.wikinews.org

⁶https://www.grimmstories.com/zh/grimm_tonghua/index

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

		Docs	Sent	Timex	Events
News	Single	91	2,271	901	3,759
	Double	24	570	266	1,048
	Total	115	2,841	1,167	4,807
Narratives	Single	96	3,034	91	9,024
	Double	24	628	40	1,952
	Total	120	3,662	131	10,976

Table 4: Corpus annotation statistics. (*Timex* stands for time expressions.)

recognition, unlabeled relations, and labeled relations respectively on a narrative data. We argue that the comparable or better agreements on narratives as shown in Table 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 5: Inter-Annotator Agreement F scores on 20% of the annotations.

4.2. 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 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 0.01%/99.99% (132/10314).

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 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 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 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 7: Distribution of event types.

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 8: Distribution of temporal relations.

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 branches of statives

with an eventive event as their parent. These statives serve as the event’s accompanying situations. For example, in (15) “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. 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 1 and Figure 2.

5. Conclusion

In this work, we proposed a novel approach to model temporal relations in a document – building a temporal dependency tree structure for the document. 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.

6. Bibliographical References

- Allen, J. F. (1984). Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154.
- Bethard, S., Derczynski, L., Savova, G., Pustejovsky, J., and Verhagen, M. (2015). Semeval-2015 task 6: Clinical tempeval. In *SemEval@ NAACL-HLT*, pages 806–814.
- Bethard, S., Savova, G., Chen, W.-T., Derczynski, L., Pustejovsky, J., and Verhagen, M. (2016). Semeval-2016 task 12: Clinical tempeval. *Proceedings of SemEval*, pages 1052–1062.
- 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, August. Association for Computational Linguistics.
- Bohnemeyer, J. (2009). Temporal anaphora in a tenseless language. *The expression of time in language*, pages 83–128.
- Chambers, N. and Jurafsky, D. (2008). 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.
- 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.
- Hinrichs, E. (1986). Temporal anaphora in discourses of english. *Linguistics and philosophy*, 9(1):63–82.
- 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.
- 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*.
- 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.
- Partee, B. H. (1973). Some structural analogies between tenses and pronouns in english. *The Journal of Philosophy*, 70(18):601–609.
- Partes, B. H. (1984). Nominal and temporal anaphora. *Linguistics and philosophy*, 7(3):243–286.
- 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, 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.
- Reichenbach, H. (1947). *Elements of Symbolic Logic*. The MacMillan Company, New York.
- Smith, C. S. (2003). *Modes of discourse: The local structure of texts*, volume 103. Cambridge University Press.
- 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. (2014). Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.
- 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*.
- 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, 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 Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic, June. Association for Computational Linguistics.
- Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Moszkowicz, J., and Pustejovsky, J. (2009). The TempEval Challenge: Identifying Temporal Relations in Text. *Lang Resources & Evaluation*, 43:161–179.
- 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, 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, Uppsala, Sweden, July. Association for Computational Linguistics.
- Webber, B. L. (1988). Tense as discourse anaphor. *Computational Linguistics*, 14(2):61–73.
- Wuyun, S. (2016). The influence of tense interpretation on discourse coherence - a comparison between mandarin narrative and report discourse. *Lingua*, 179:38 – 56.
- 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.
- 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.
- 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.
- Zhang, Y. and Xue, N. (2014). Automatic inference of the tense of chinese events using implicit linguistic information. In *EMNLP*, pages 1902–1911. Citeseer.