Text machine laboratory

NarrativeTime annotation guidelines
version 1.0.0

Anna Rogers, Marzena Karpinska, Anna Rumshisky

December 23, 2022

Contents

1 What constitutes an event?

Any annotation of temporal order necessarily relies on some definition of events, between which the order is established. NarrativeTime framework for event order annotation can be used with events defined in any way, but for the purposes of illustration in the current guidelines we adopt the TimeML definition of an event as anything that happens or occurs in the world of the narrative. It may be an individual or a repeating event, a process or a state. Syntactically, it may be expressed in several ways:

1. Verbs and predicative clauses:
   - John [**kissed**] Mary.
   - John was [**running**] for half an hour.
   - John was [**interested**] in politics.
2. Nouns or noun phrases:
   - John was [**looking forward**] to the [**dinner**].
   - John was [**invited**] to the [**shareholder meeting**].
   - John was a [**soldier**].
3. Adjectives and participles:
   - John [**saw**] the [**startled**] girl.
   - John was [**there**], [**looking**] at Mary with much interest.
   - John was [**tall**] and [**hard to ignore**].

Temporal annotation is generally performed in 2 stages: (1) identification of events (and, optionally, their coreference), (2) establishing the temporal order of these events. NarrativeTime focuses on event order. Events can be marked up using any existing event markup schemes such as ECB+, TimeML, RED or other. Different schemes have different policies about what gets annotated and how (especially with regard to phrasal verbs, support verbs, modal phrases etc.)

The current guidelines assume that the annotation of events themselves has already been performed,[1] with one of the forementioned definitions, and focus exclusively on their order. All examples show which events are marked as [**events**]; different event markup on the same examples could result in different NT annotation.

Note also that most annotation efforts do not annotate everything that *could* count as an event, and as you work with temporal order you may see events that are unmarked. In this case, you will need to either mark them and re-import the file, or proceed with only the annotated events as targets.

[1]The NT annotation tool does support events annotation, which can be performed according to the guidelines of one's choice.

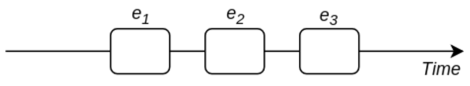
2 Timeline representation of event order

When you read a story, you create a rough mental representation of what happened, and in what order it happened. This representation can be thought of as a timeline. It is such a timeline that you partly remember once you have read a story and are asked to retell the plot to someone else.

The goal of NarrativeTime is to create a complete representation of the temporal order of all events in a given narrative. The easiest way to represent event order graphically is a timeline. Consider a simple example:

(1) John [**met**]^{e_1} Fred, [**ate**]^{e_2} a sandwich, and [**watched**]^{e_3} a movie.

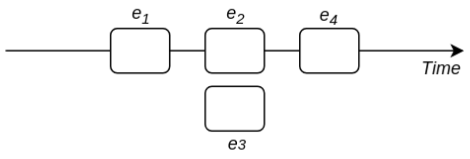
The three events in this mini-story happen consecutively, and their order could be represented schematically as follows:



For this kind of representation, we need only two kinds of temporal relations: the before/after relation (2 events are successive on the timeline), and the simultaneous relation (2 events occupy the same position on the timeline).

Ex. 2 shows both consecutive and simultaneous events, and the mechanism for achieving this representation:

(2) John and Fred [**entered**]^{e_1} the diner. John [**ate**]^{e_2} a sandwich, and Fred [**ate**]^{e_3} a toast. Then they [**talked**]^{e_4} about football.



Event	Timeline position
John and Fred [**entered**]^{e_1} the diner.	1
John [**ate**]^{e_2} a sandwich	2
Fred [**ate**]^{e_3} a toast.	2
Then they [**talked**]^{e_4} about football.	3

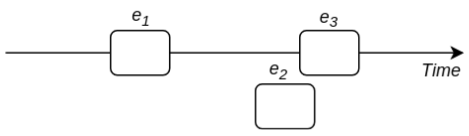
The table below the timeline representation in ex. 2 shows the basic mechanism by which the timeline representation is constructed in the NarrativeTime annotation tool (see chapter 4). Each event is associated with a number that indicates its temporal position. Smaller numbers indicating temporal precedence before the bigger numbers, and equal numbers stand for simultaneity. Based on these numbers, an interactive timeline representation is automatically updated.

In this example, an event at position 2 precedes the event at position 3 and follows the event at position 1. We actually have two events at position 2: they are considered to be simultaneous.

The numbers do not have to be integers: for example, if you have already created events at positions 2 and 3, and then another event comes up that should have been placed between them, you can attribute it to the position 2.2, 2.5 etc. As long as the order of the numbers is correct, the timeline representation will be correct. Zero and negative numbers are also allowed (the latter are particularly useful for stories with flashbacks).

2.1 Roughly-consecutive events

Most narratives are not as simple as ex. 1 or ex. 2, and the temporal overlaps are not always neat. Even in ex. 1, it is theoretically possible that John was still finishing his sandwich while already watching the movie, and in that case a more precise representation for ex. 1 would be as follows:



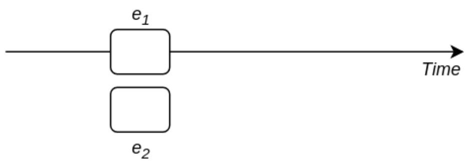
However, in the real life we rarely time events with stopwatches, and normal process of reading or listening we are not concerned with such details: it is sufficient that we have a rough idea of the order of events. NarrativeTime does not ask you to do anything more than that. Events may be roughly-consecutive even if there is minor temporal overlap: it is sufficient that the bulk of the duration time of the two events is consecutive.

2.2 Roughly-simultaneuous events and subevents

In NarrativeTime events are allowed to be roughly-simultaneous rather than strictly-simultaneous. This enables us to attribute the following events to roughly the same temporal position:

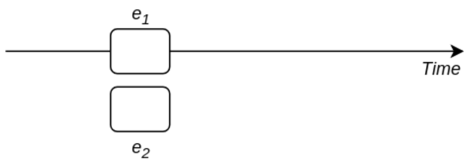
(3) John [**talked**]^{e_1} to Mary at [**breakfast**]^{e_2} in Starbucks.

In this example, it is possible that John bumped into Mary while walking into Starbucks, or while he was walking out of it, or at any other point between these two. However, the author of the story does not provide this information directly, which indicates that it is not crucial to the story. So a simple, roughly-simultaneous interpretation suffices:



The same logic applies to representation of sub-events. In many cases the temporal order information is not sufficiently specified, and the temporal order imprecision between super- and sub-events are not salient enough to notice, unless you start over-thinking it.

(4) John had [**breakfast**]^{e_1} at Starbucks. The coffee [**smelled**]^{e_2} great.

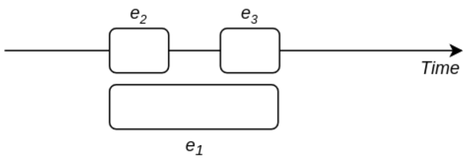


In this example, coffee is probably only a part of John's breakfast, and we do not know whether he started drinking the coffee while he had his toast, before, or after that. But it does not really matter. The writer does not focus our attention on anything else that happened in the potential gaps, so these two events can be considered roughly-simultaneous.

What the above examples have in common is **the absence of other narrated events that could fit in the duration of the longer event**. NarrativeTime assumes that if the temporal gaps between the start/end of sub-event and the super-event are salient in the context, it will be made clear by the presence of other events that together occupy the duration of the super-event. For example, the same coffee/breakfast situation could be viewed and represented differently simply if another sub-event is mentioned, which forces the coffee towards the latter part of the "breakfast" super-event:

(5) John had [**breakfast**]^{e_1} at Starbucks. He [**paid**]^{e_2} with his credit card. The coffee [**smelled**]^{e_3} great.

In this case, “payment” and “coffee” are consecutive sub-events spanned by the “breakfast” super-event:



Depending on your perspective, you could also interpret ex. 2.2 as e_1 and e_3 overlapping, and e_2 preceding them. That annotation would construe the payment as being separate from the breakfast activity. Your annotation should reflect your intuitive understanding of the situation, there is no need to over-think anything.

2.3 Temporal gaps between events

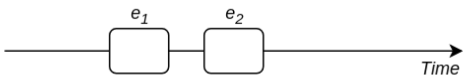
Two consecutive events may be immediately adjacent (the second event starts when the first event ends), or there may be an interval between them.

(6) John [**fell down**]^{e_1} and [**broke**]^{e_2} his leg.

(7) John [**woke up**]^{e_1} and [**brushed**]^{e_2} his teeth.

In the first example, breaking the leg happened immediately after the falling event. In the second example the temporal gap is underspecified: it is possible that John brushed his teeth immediately upon waking up, but he also could have showered in the interval between these two events.

NarrativeTime does not make this distinction, since that would have to rely on commonsense knowledge rather than the information in the text. Both examples would have the following timeline representation in NarrativeTime:



This approach means that the exact intervals between numbers that serve as temporal positions for the events in the annotation interface do not matter. Events located at 1 and 2 are considered to be in the same kind of relation as events located at 1.1 and 1.102, or 1 and 100: they are consecutive, as long as there are no other events between them.

3 Types of events in NarrativeTime

Few events in texts are fully specified with respect to their temporal position; in the normal reading process the reader infers much of it, and the resulting representation is not always 100% the same as what was intended by the writer. NarrativeTime incorporates the underspecification in the definitions of annotation units, so that the annotators do **not** have to over-think temporal relations that are not known. There are three basic types of events.

1. **Bounded events** (section 3.2): there are other events in the context that constitute temporal "boundaries" for the given event, i.e. it is clear what happens before and after that event.
2. **Unbounded events** (section 3.3): the context provides no events that clearly happen before/after the given event.
3. **Partially bounded events** (section 3.4): there is an event that clearly happens before the given event, but it is not clear when it ends (or vice versa, it is clear when the event ends but not when it starts).

Thus the type of event in NarrativeTime depends primarily on the context, which either has or does not have other events that clearly precede/follow the given event. Consider the following examples:

(8) a. John [**received**] his degree. He did 2 years in the [**army**]. He got [**married**].
b. John [**received**] his degree. He joined the [**army**]. He got [**married**].

The set of John's life events that we learn from ex. 8 (a) and (b) is similar, but their timeline representation is not the same. In 8 (a) the army event is fully **bounded**[1]: the consecutive order of the three events would lead a typical reader to believe that they did not overlap. In this case, the degree event and marriage event provide temporal boundaries for the army event. In 8 (b), the army event has only one temporal "boundary": the degree. We no longer know when the army event ends, but the marriage event clearly happens during the army service. It is only **partially bounded**.

This means that in NarrativeTime the same kinds of events may be bounded in some contexts and unbounded in others: what matters is whether in this given context there are other events that constitute their temporal "boundaries". It does not matter how long a bounded event lasts, or whether it is a short event or a process. As long as it is clear what other event precedes/follows a given event `X` on the timeline, the event `X` is bounded. In 8 (b) the army could become a bounded event if at a later point the text said "after John left the army, he joined the police."

The following sections describe each type of event in detail, but the overall idea is summarized in Figure 3.1. With practice, to determine the type of any given event, you will need to answer no more than two simple questions.

Since to determine the type of event we need to know the preceding and following events, the first and the last events in the story are tricky: we simply are not told what comes before/after them. **The NarrativeTime convention is to always treat the beginning and the ending of the text as a boundary**. The first/last events can thus be either bounded or partially bounded (see section 3.4).

3.1 Events vs spans.

We could annotate event order by working on single events one by one, but in many cases this is not necessary, because several events may be roughly-simultaneous or roughly-consecutive. We refer to such clusters of events as **spans**, because during annotation you can create such annotations by highlighting a continuous span of text. From now on we will show suggested spans by [highlights] in the body of examples. For example, 8 (a) could be annotated with three spans or a single span, as shown in 9. We will discuss further when it is possible to save annotation effort in this way.

[1]We use the term "boundedness" primarily with respect to contextual construal of an event rather than Aktionsart properties of a verb. Classical bounded events such as "eating a cake" tend to be bounded in a narrative, i.e. their temporal position will be clear with respect to previous and following events, but they could also be viewed as a process.

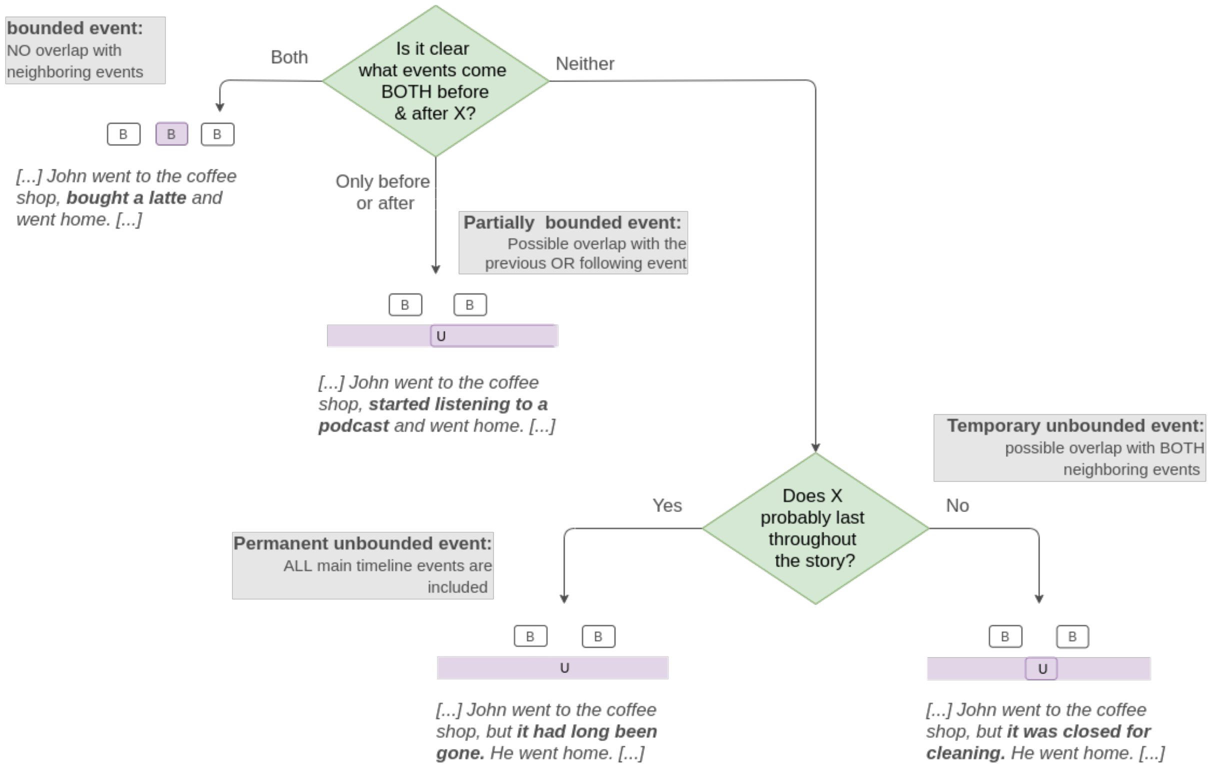

bounded event:
NO overlap with neighboring events
Both
Is it clear what events come BOTH before & after X?
Neither
B
B
B
[...] John went to the coffee shop, **bought a latte** and went home. [...]
Only before or after
Partially bounded event:
Possible overlap with the previous OR following event
B
B
U
[...] John went to the coffee shop, **started listening to a podcast** and went home. [...]
Temporary unbounded event:
possible overlap with BOTH neighboring events
Yes
Does X probably last throughout the story?
No
Permanent unbounded event:
ALL main timeline events are included
B
B
U
[...] John went to the coffee shop, but **it had long been gone.** He went home. [...]
B
B
U
[...] John went to the coffee shop, but **it was closed for cleaning.** He went home. [...]

Figure 3.1: Determining the type of event in NarrativeTime.

(9) a. [John [**received**] his degree. He did 2 years in the [**army**]. He got [**married**].]
b. [John [**received**] his degree.] [He did 2 years in the [**army**].] [He got [**married**].]

Since it is the spans that we actually create in the annotation interface, the event types (bounded/unbounded) are actually also the span types. All events within a bounded span are considered bounded, and all events within an unbounded span are considered unbounded. With one exception (§3.2.3), where the events are (roughly-)consecutive, all events within a given span also have the same position on the timeline.

3.2 Bounded events

Bounded events correspond to the button **[B]** in the annotation interface. "B" is also the shortcut for creating a span of this type, which contains either a single event for which we know what happens before and after it (§3.2.1), or several such events that are roughly-simultaneous (§3.2.2). In §3.2.3 we will also discuss the special case of a consecutive chain of bounded events, which often occurs in narrative texts.

3.2.1 Single events or processes with known start/end points

A succession of events that clearly precede/follow each other is the most straightforward case for temporal annotation. Each event in such a chain could be viewed as a separate bounded event.

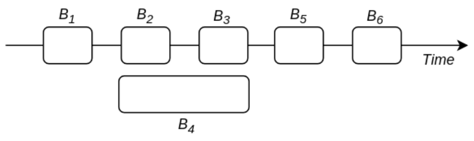
(10) John did 2 years in the[[**army**]]$_B$, [[**received**] his degree]$_B$, and [[**married**]]$_B$ Mary.

In this case, army and marriage events are the boundaries for the degree event. Since this story is so short, this is all we know about John, and so we cannot tell what preceded the army and followed the marriage events. But these are the start/end events, and so they get the missing boundaries by the NarrativeTime convention.

It is possible for one [B] span to be a "super-event" which covers several consecutive "sub-events" (see 11), as long as the start/end points roughly coincide:

(11) [John, Fred and Mary [**entered**] the coffee shop]$_B$^{1}. [John [**ordered**] a coffee for Mary]$_B$^{2}, and [she [**drank**] it]$_B$^{3}. [Fred [**leafed**] through a newspaper]$_B$^{4}. [They [**got up**]]$_B$^{5} and [[**left**]]$_B$^{6}.

Assuming that the first and the last event in this story are also bounded by the rest of the context, it can be represented as follows, with numerical interval `x:y` indicating the start and end points of the spanning event:



Event		Timeline position
John, Fred and Mary [**entered**] the coffee shop	[**B**]	1
John [**ordered**] a coffee for Mary	[**B**]	2
she [**drank**] it	[**B**]	3
Fred [**leafed**] through a newspaper	[**B**]	2:3
They [**got up**]	[**B**]	4
and [**left**]	[**B**]	5

This annotation means that Fred's reading activity goes on for the combined duration of "John ordered a coffee for Mary", and "Mary drank it." Both sequences are bounded by entering and exiting the coffee shop. As discussed in chapter 2, we do not assume that the event sequence is precise (maybe Fred finished the newspaper a little before Mary finished coffee). During annotation, you can read the way you usually read, without overthinking each event boundary.

Note: you are annotating only the actual event(s) mentioned in the text, and not the consequent state (unless that state is mentioned separately and can be annotated in its own right). For example, consider the span:

(12) [John [**married**] Mary.]$_B$ [They [**went**] to New Zealand together.]$_B$

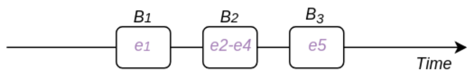
It could be argued that John remains married during the New Zealand trip. However, it is the event of getting married that is explicitly mentioned, not the subsequent state of being married. And so we treat it as a bounded event.

3.2.2 Spans of roughly-simultaneous events with known start/end points.

A single [**B**] span may also include roughly simultaneous events, the order of which is not clear and/or not important for the purposes of the current narrative, and which occupy the same continuous chunk of time. Bundling them together as a single [**B**] event span encodes them as a roughly simultaneous event, saving annotation effort.

(13) [John and Mary [**broke up**]^{e_1}.]$_B$^{1} [He incessantly [**texted**]^{e_2}, [**called**]^{e_3}, and [**left**]^{e_4} voicemails for her.]$_B$^{2} [She finally [**changed**]^{e_5} her phone number.]$_B$^{3}

In this example, the precise order of events e_2-e_4 is neither known nor important, and so a more concise representation can be achieved by means of putting them in one [**B**] span.



This mechanism may be used even when the characters and/or place of action are different:

(14) <..> [Everything [**went on**]^{e_1} as usual: the teachers [**taught**]^{e_2}, the students [**yawned**]^{e_3}.]$_B$ <..>

3.2.3 Spans of consecutive events with known start/end points

You saw above how to save annotation effort with encoding several roughly-simultaneous events as a single **[B]** event. What about sequences of **[B]** events? It is also a very frequent narrative structure to list the events in the order in which they happened. Do they all have to be annotated separately?

Since annotation of such narratives would involve simply creating spans for all these steps of the narrative in the order in which they come in the text, we have a special type of event span **[C]** (for "consecutive") that serves as a shorthand for several **[B]**...**[B]** spans. The following annotations are equivalent:

(15) a. [John did 2 years in the [**army**]]$_B$^{1}, [[**received**] his degree]$_B$^{2}, and [[**married**] Mary.]$_B$^{3}

b. [John did 2 years in the [**army**], [**received**] his degree, and [**married**] Mary.]$_C$^{1}

You can create such spans of any length, as long as:

(1) the events are all bounded, in consecutive order and with the same factuality status (see **??**;

(2) there are no explicit temporal expressions (see chapter 6);

(3) there are no events that serve as anchor points for branches (see chapter 5, or for some other event that is simultaneous with only a part of the sequence.

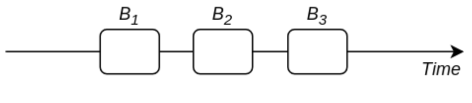
Conditions (1-2) are immediately clear from the tentative event sequence themselves, but (3) is trickier because it depends on the rest of the text. For example, you may sometimes find yourself creating a **[C]** span, and then finding that you need one of the events to be separate because it is simultaneous or a starting point for something else. In such cases simply re-create the annotation, simply splitting the **[C]** span into as many **[B]** spans (or **[C]** and **[B]** span combinations) as necessary.

3.3 Unbounded events

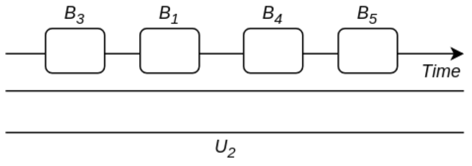
The event type complementary to **[B]** is **{U}**, standing for "unbounded" event spans. Remember that the defining feature of a **[B]** event is that it has clear temporal boundaries set by other events in the context: it ends before a certain other event, and it starts after a certain previous event. An event is unbounded simply when such clarity is lacking, in either (see §3.4) or both directions. A **{U}** event is overlapping with the other neighboring events, but the exact boundary is not clear. All we do know is when the ongoing phase of **{U}** event is taking place.

An event may thus be bounded or unbounded not because of the lexical properties of the target word, but because of the context. Ex 16 (a) shows that even a process like "work" could be attributed to **[B]** type, if it is presented as **an event with a clear position on the overall timeline.**

(16) a. [John [**graduated**] in 2010]$_B$^{1}, and [then he [**worked**] as an accountant]$_B$^{2}. [He then [**started**] his own company]$_B$^{3}.



b. [John [**bought**] a gym membership]$_B$^{1}. {He [**worked**] as an accountant}$_U$^{2}, but [after he [**trained**] for two months]$_B$^{3}, [he [**lost**] weight]$_B$^{4} and even [[**ran**] a marathon]$_B$^{5}.



In both ex. 16 (a) and (b) John works as an accountant, but in (b) we do not know when this starts or ends. We do know that he had that job throughout the gym membership and training events, i.e. it definitely overlaps with these events and maybe also some others in the context. Note that the order of B_1 and B_3 is not the narrative order. The world knowledge suggests that people first buy the gym membership and then start training; the former enables the latter.

3.3.1 "Permanent" and "temporary" unbounded events

The defining feature of **{U}** events is that the start and end points are not known. However, that definition alone applies equally to the following two examples:

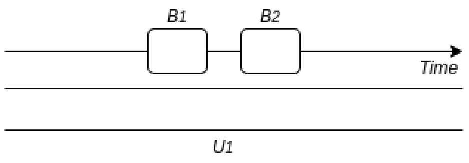
(17) a. [Mary [**came in**]]$_B$^{1}. {John was [**fat**], not at all like his profile picture.}$_U$^{1} [She [**said**] hi.]$_B$^{2}
b. [Mary [**came in**]]$_B$^{1}. {John was [**watching**] TV.}$_U$^{1} [She [**said**] hi.]$_B$^{2}.

In both cases, it is not clear when exactly the middle **{U}** state started and when it ends. However, we can be sure that the first lasted much longer than the second, and it would be misleading to have them represented on the timeline in the same way.

To account for this difference, NarrativeTime distinguishes between **"permanent"** and **"temporary" U spans**. In the annotation interface, they are created with the same **{U}**, but the temporary spans have a specific temporal position, just like **[B]** events (e.g. "John was watching TV" around timeline position 5). The permanent spans do not. If you need to create a permanent span, simply replace the number indicating its timeline position in the annotation table with ":", or just delete it (and ":" will be added automatically).

"Permanent" event spans are defined when the event (typically, an attribute, a state, or a process) is **probably lasts for the duration of the story**. Typical examples include traits of character and descriptions of appearance. In 17 (a) John is fat; it is possible that he will join the gym and lose weight, but this is not a naturally expected change, and if they were to happen and be important for the story, we would probably be told about it. On a timeline, we could represent this **{U}** state as follows:

[Mary [**came in**]]$_B$^{1}. {John was [**fat**], not at all like his profile picture.}$_U$^{1} [She [**said**] hi.]$_B$^{2}



If John does lose weight later, you will need to change the type of this span to the partially bounded event **{U]**, ending at the point at which he loses weight (to be discussed in §3.4).

"Permanent" U spans are also used for descriptions of objects; e.g. if a table is white, it is not impossible that it will get painted green, but without being told about that explicitly we may assume it stayed white throughout the story. If we are subsequently told that it was painted green, the whiteness becomes a partially bounded event (see section 3.4).

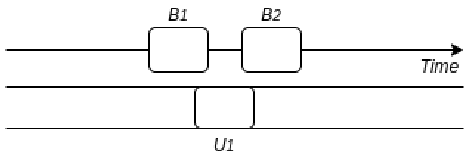
Generic, habitual, or normally-recurring events (i.e., events that are presented as occurring often/all the time) are also described as "permanent" U spans in NarrativeTime. The rationale is the same: the truth of generic statements applies equally throughout the timeline of the current narrative, or even beyond it.

(18) a. {Everybody [**lies**]}$_U$.
b. {John was often [**wearing**] jeans and t-shirts at that time}$_U$.
c. {Tigers sometimes [**catch**] caimans}$_U$.

You may remember that normally, events are associated with a particular number that indicates their temporal position on the timeline. Since the "permanent" spans by definition are not associated with a particular temporal position, in the annotation interface their temporal position is specified as ":", and not as a number. This is a shortcut equivalent to duration specified as x:y, where x is the first point on the timeline and y is the last point on the timeline.

"Temporary" event spans. This is the type of span for events that are not clearly bounded by the neighboring events, but also probably do not hold throughout the whole story. In the example 17 (b) we have a temporary **{U}** span, because it is possible that John kept watching TV as Mary was walking in, and maybe even while she was greeting him (and only then he would turn to her). He probably does not keep watching it forever, but he was definitely watching it as she entered: this is the temporal "center" of this unbounded event.

[Mary [**came in**]]$_{B}{}^{1}$. {John was [**watching**] TV.}$_{U}{}^{1}$ [She [**said**] hi.]$_{B}{}^{2}$.



3.3.2 Spans of unbounded events

As described above, you can save annotation effort by creating a single [**B**] that includes several roughly-simultaneous events:

(19) <..> [Everybody [**went about**] their business as usual: the teachers [**taught**], the students [**yawned**].]$_{B}$ <..>

The same can be done with unbounded spans: if you have several **{U}** processes that occupy the same time frame (either both "permanent", or with the same temporal "center"), you may group them together in a single **{U}** span. The following annotations are equivalent, provided that both spans in 20 (a) are attributed to the same temporal position on the timeline.

(20) a. <...> {The teachers were [**teaching**]}$_{U}{}^{1}$, {the students were [**yawning**]}$_{U}{}^{2}$. <...>
b. <...> {The teachers were [**teaching**], the students were [**yawning**]}$_{U}{}^{1}$. <...>

In some cases generic events of **{U}** type may also form narratives with temporal structure. For instance, the following could be considered as four generic events that do occur in that specific order:

(21) {People are [**born**]}$_{U}$, {people [**grow up**]}$_{U}$, {people get [**old**]}$_{U}$, {people [**die**]}$_{U}$.

However, technically, they all "always" take place. At the moment, the policy is to simply consider them all to the "permanent" **{U}** process, so the following annotation would also be correct:

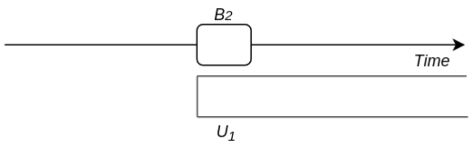
(22) {People are [**born**], people [**grow up**], people get [**old**], people [**die**]}$_{U}$.

As you remember, the generic, always-true events are exempt from the rule that the start/end of the story constitutes a temporal boundary. So even this standalone example can be annotated as a "permanent" unbounded event.

3.4 Partially bounded events

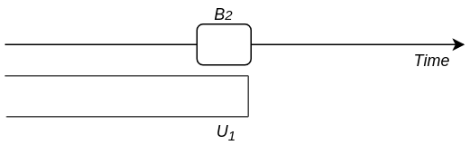
So far we have events that either have clear boundaries both "before" and "after" (the bounded [**B**] events), or neither of these boundaries is clear (the unbounded **{U}** events). The only remaining case is the "partially bounded" events: as you can guess, this means that only one of the temporal boundaries is clear. We know when such event starts or ends, but not both. They are marked as **{U]** and **[U}**, with the curly bracket pointing to the uncertain part of the timeline. For example:

(23) a. [Mary [**loved**] jazz}$_{U}{}^{1}$ since [[**high school**].]$_{B}{}^{2}$



Event		Timeline position
Mary [**loved**] jazz	**[U}**	1
high school	**[B]**	1

b. {Mary [**loved**] jazz]$_U$^{1} until [she [**met**] John]$_B$^{2}.



Event		Timeline position
Mary [**loved**] jazz	**{U]**	1
she [**met**] John	**[B]**	1

To annotate partially bounded events we need an “anchor”: the temporal position of the point at which the **{U}**-event started or ended. That position is simply the position of the event that bounds the **{U}** event in either direction. In the examples above, the partially bounded events are anchored at position 1. Since the starting/ending points of the story are also “boundaries”, the very first event may also be a **[U}**, and the very last one - a **{U]**.

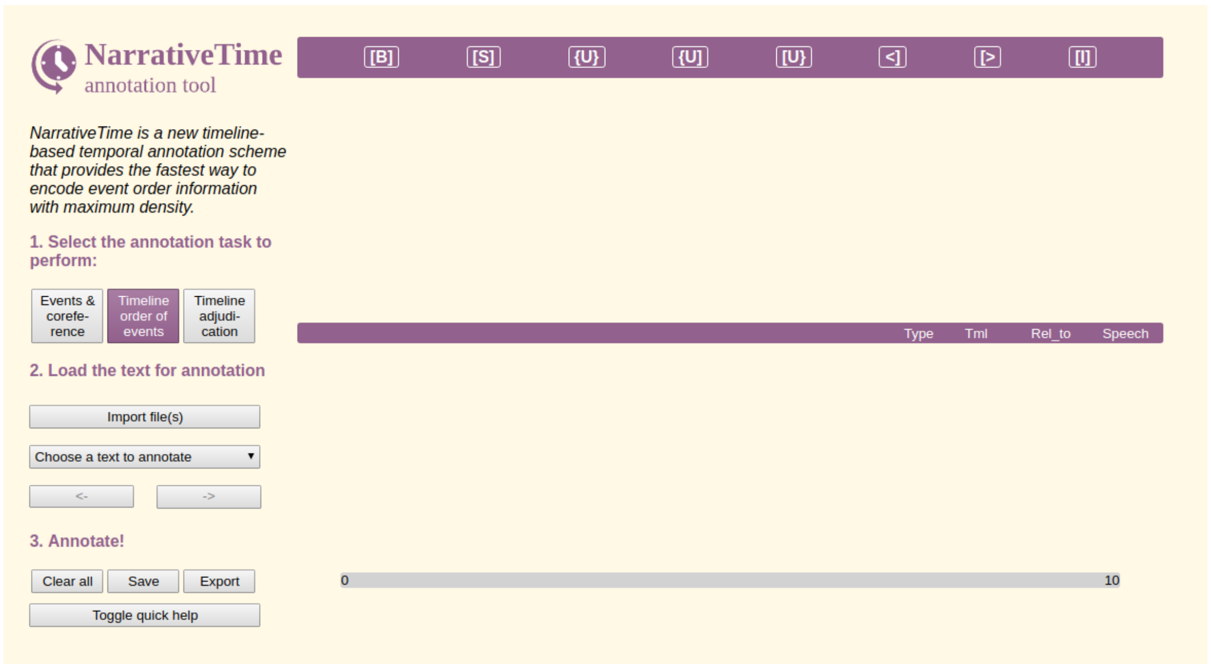
4 NarrativeTime annotation tool

So, how do we use all these event types to quickly annotate event order? NarrativeTime has a specialized web-based annotation tool. No installation is necessary. We recommend using Chrome or Chromium browser, as other browsers may not display all the elements correctly.

The interface has three major elements:

- the control panel that provides general annotation process controls such as saving your work and navigating between texts;
- the main annotation panel that contains the text and timeline annotations;
- a quick help panel with brief guidelines, which can be toggled from the control panel.

[**TO DO: change the screenshots. –MK**]



NarrativeTime is a new timeline-based temporal annotation scheme that provides the fastest way to encode event order information with maximum density.

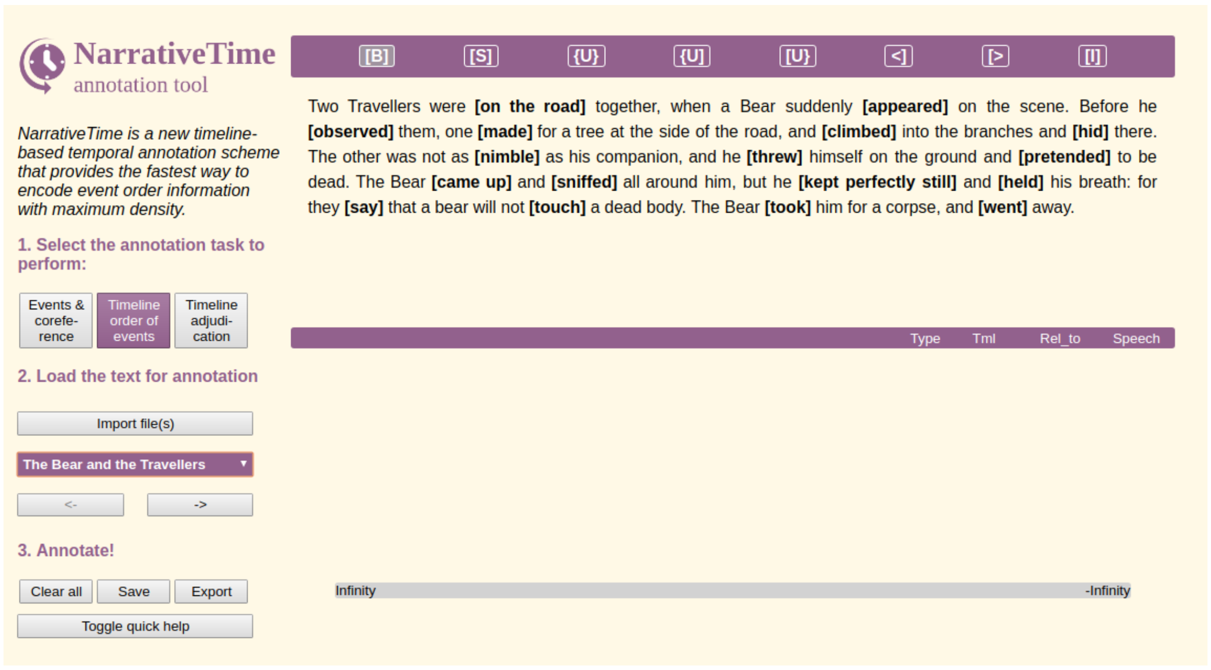
Figure 4.1: The starter screen

In the offline version of the tool, you will need to manually load a .jsonl file for annotation. Saving is done in two steps: the "save" button saves your work in the browser cache (and will be lost if you reload the page, restart the browser, or change the annotation type). The "export" button lets you save it to the disk as a new .jsonl file. This can be done just once, when you have finished a work session for the day. An exported .jsonl annotation file can be re-imported to view and change annotations.

The same software is used for the 3 stages of the annotation process (even markup and coreference, timeline order, and adjudication), and there are buttons for switching between these interfaces. Depending on the task at hand, they may be disabled by the administrator to prevent editing at the wrong annotation levels.

The navigation between texts is performed with either text title selector from the list, or the navigation buttons under the text selector.

Once a specific text is selected for annotation, it will be displayed in the annotation area to the right of the control panel. At this stage, the texts should already contain the annotation for events, event coreference and temporal expressions. When you load a text for annotation, it should look roughly as follows:



NarrativeTime is a new timeline-based temporal annotation scheme that provides the fastest way to encode event order information with maximum density.

Two Travellers were **[on the road]** together, when a Bear suddenly **[appeared]** on the scene. Before he **[observed]** them, one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there. The other was not as **[nimble]** as his companion, and he **[threw]** himself on the ground and **[pretended]** to be dead. The Bear **[came up]** and **[sniffed]** all around him, but he **[kept perfectly still]** and **[held]** his breath: for they **[say]** that a bear will not **[touch]** a dead body. The Bear **[took]** him for a corpse, and **[went]** away.

Figure 4.2: Text loaded for annotation

4.1 Working with the annotation interface

The following story will be the first text to be annotated in the current pilot annotation round. Use it as a training example, working with it as you read the subsequent sections of the guidelines:

> Two Travellers were [**on the road**] together, when a Bear suddenly [**appeared**] on the scene. Before he [**observed**] them, one [**made**] for a tree at the side of the road, and [**climbed**] into the branches and [**hid**] there. The other was not as [**nimble**] as his companion, and he [**threw**] himself on the ground and [**pretended**] to be dead. The Bear [**came up**] and [**sniffed**] all around him, but he [**kept perfectly still**] and [**held**] his breath: for they [**say**] that a bear will not [**touch**] a dead body. The Bear [**took**] him for a corpse, and [**went**] away.

All styled events must be taken into account while performing temporal annotation. Anything else can be ignored. For instance, if an event is mentioned several times, only one of these mentions will be pre-selected for placing on the timeline.

4.1.1 Creating event spans

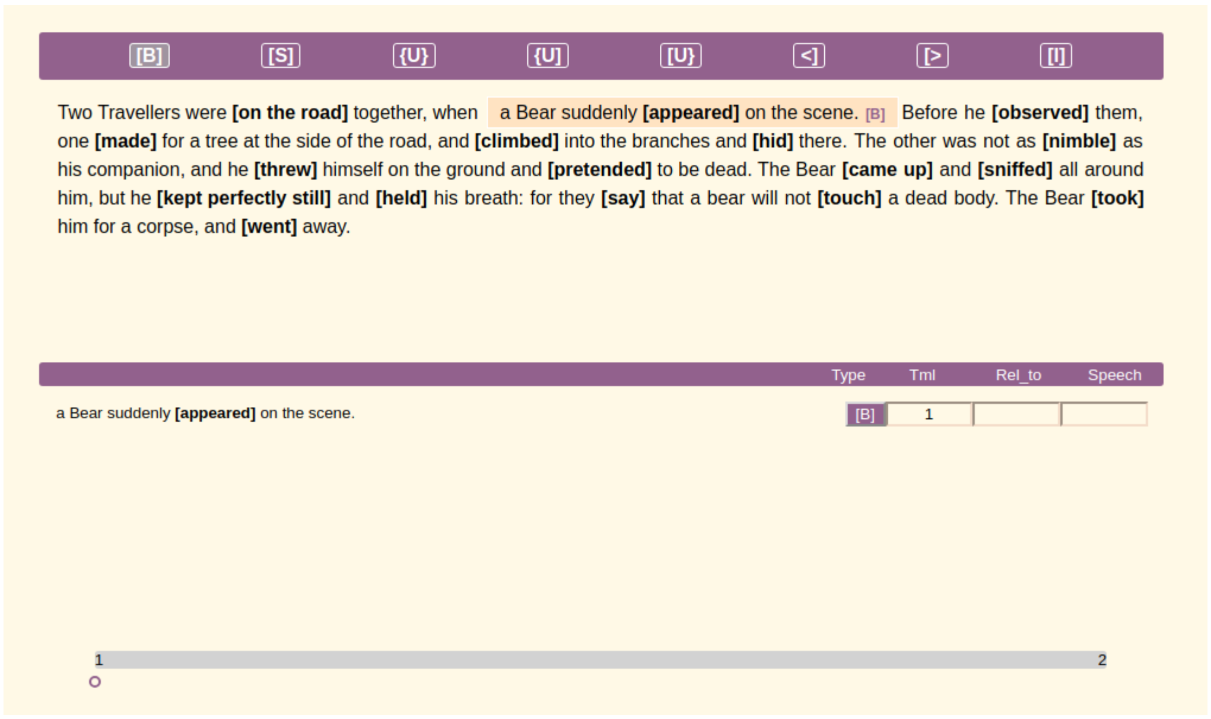
To perform annotation of event clusters in the web interface, simply (a) select the correct type of the event with a mouse or shortcut, and (b) highlight the relevant text spans. As you highlight the spans, they will appear in the table under the text box. As described above, the buttons for types of spans include **[B]**, **[C]**, **{U}**, **{U]**, **[U}**, and **[I]**. In addition to that, there are two extra buttons that serve as shortcuts for creating branching timelines (to be discussed in chapter 5.

The shortcuts correspond to the type labels - *b*, *s*, *u*, *i*. In case of multiple label per key, the shortkeys cycle through all the appropriate labels, e.g. "U" key will cycle through **{U}**, **{U]**, **[U}**.

Let us start by selecting the first clear bounded event:

The span is now highlighted, and under the text there is a table with the text spans, span types, and 3 input fields. The button in the Type column corresponds to the type of the event that you selected. If you need to change it, you can also do so by clicking on this button, or (once it has been clicked) by using the SPACE key.

The input fields are used as follows:



Two Travellers were **[on the road]** together, when a Bear suddenly **[appeared]** on the scene. [B] Before he **[observed]** them, one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there. The other was not as **[nimble]** as his companion, and he **[threw]** himself on the ground and **[pretended]** to be dead. The Bear **[came up]** and **[sniffed]** all around him, but he **[kept perfectly still]** and **[held]** his breath: for they **[say]** that a bear will not **[touch]** a dead body. The Bear **[took]** him for a corpse, and **[went]** away.

	Type	Tml	Rel_to	Speech
a Bear suddenly **[appeared]** on the scene.	[B]	1		

Figure 4.3: Creating the first span.

- The **Tml (timeline)** field serves for inputting single numbers (e.g. 1) or number ranges (e.g. 1:3) that indicate temporal position and duration of an event. This field is auto-populated by integers in the consecutive order as you select spans, so that if events happen in the order 1, 2, 3, and you select them in that order, then the correct timeline positions will be inserted automatically. You can also edit them manually. //

  If you need to link discontinuous spans that refer to the same event or sequence of events, this can be done by appending extra characters to mark which spans should be joined (see section 4.2).

- The **Rel_to ("relative to")** field is used for indicating events with relative temporal positions that are not on the main timeline of the story (see chapter 5). In most cases only the main timeline is used, so this field can be left empty.

- The **"Speech" field** is used for marking direct and indirect character speech, as well as temporal position of implicit speech events (see **??**).

Any change in annotation (creation, editing or deletion of a span in the annotation table) automatically refreshes the timeline representation of the narrative. Currently we have only one event, so it is shown in the beginning of the timeline at the temporal position 0.

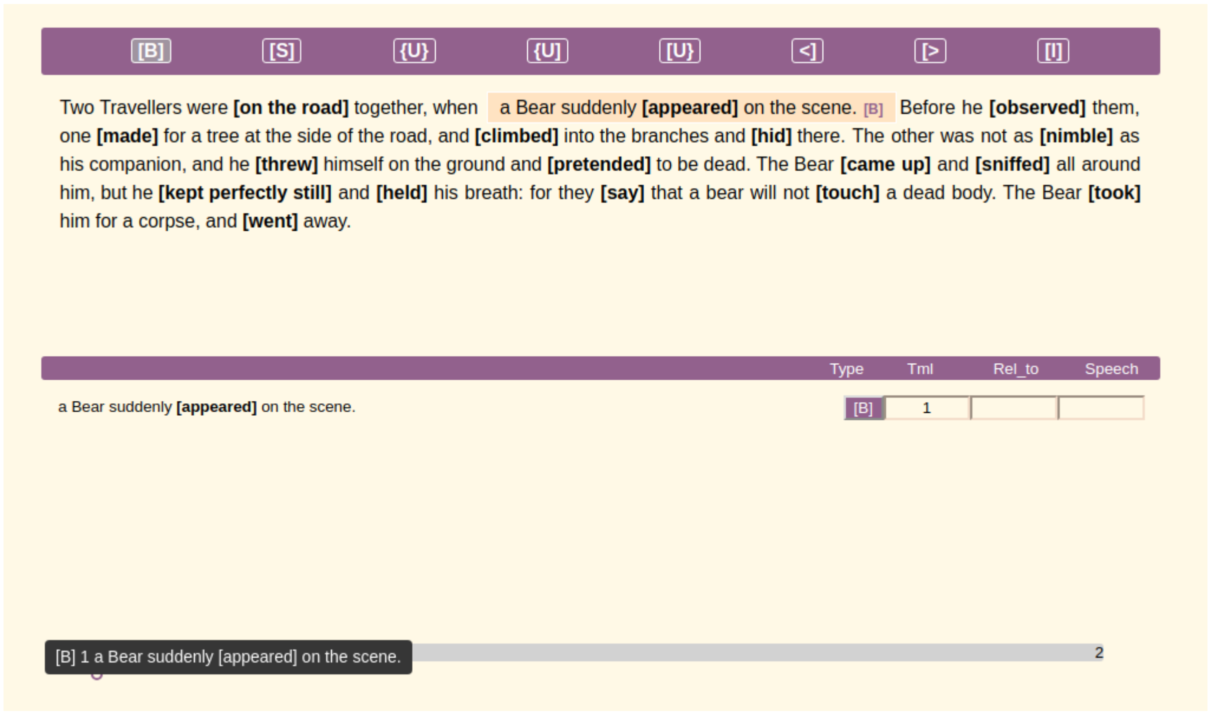
Hovering over any element of the timeline will bring up a tooltip with the text of the span, as showin in **??**. This provides a a convenient bird's-eye view of the story.

Clicking any element of the timeline will (a) highlight the corresponding span in the text and in the annotation table, and (b) activate the timeline position field - so that you could quickly check the context and/or edit the span position if need be. This functionality is shown in **??**

To delete an annotation, simply click again on the corresponding span. To delete all annotations and start over, use the "Clear all" button in the left-side control panel.

4.1.2 Ordering event spans

The numbers on the right of the span table specify the temporal order on the timeline. They will appear in the order in which they are highlighted. So if you highlight spans in the order in which the corresponding events occur, you will save time and effort on reordering them manually. For example,

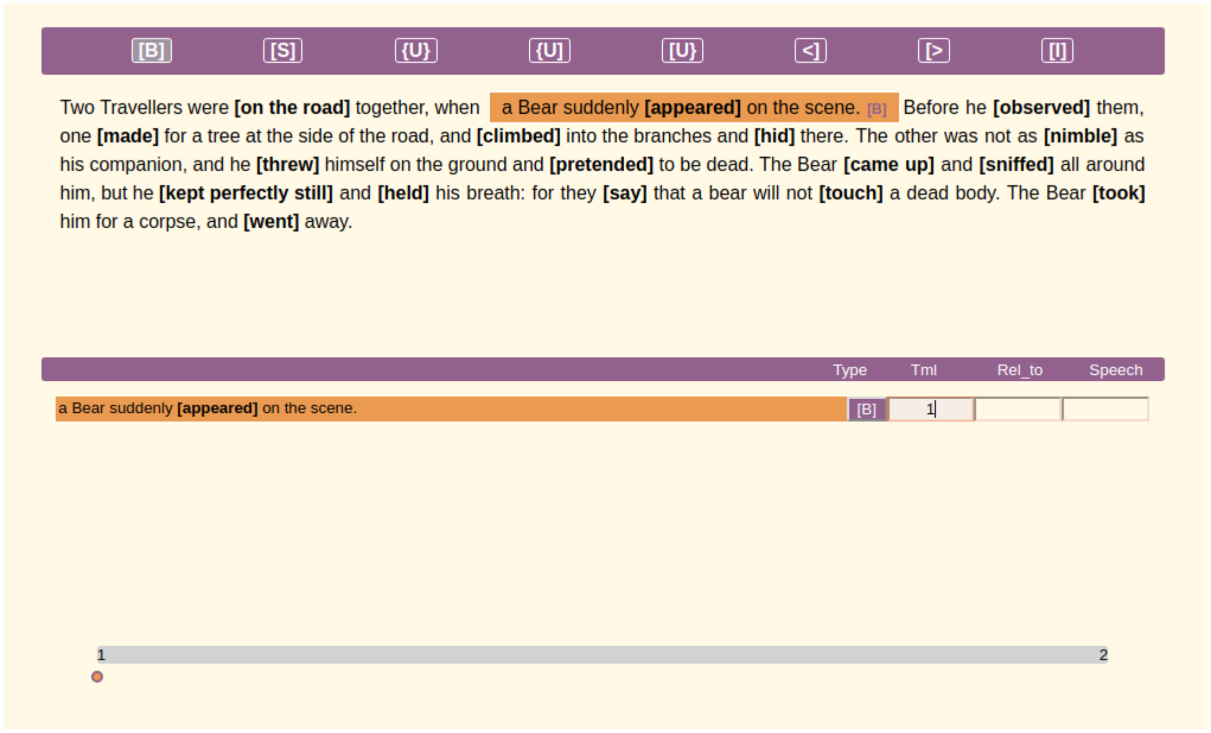


Two Travellers were **[on the road]** together, when a Bear suddenly **[appeared]** on the scene. [B] Before he **[observed]** them, one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there. The other was not as **[nimble]** as his companion, and he **[threw]** himself on the ground and **[pretended]** to be dead. The Bear **[came up]** and **[sniffed]** all around him, but he **[kept perfectly still]** and **[held]** his breath: for they **[say]** that a bear will not **[touch]** a dead body. The Bear **[took]** him for a corpse, and **[went]** away.

a Bear suddenly **[appeared]** on the scene.

[B] 1 a Bear suddenly [appeared] on the scene.

Figure 4.4: Viewing the text associated with a span element by hovering over it on the timeline.



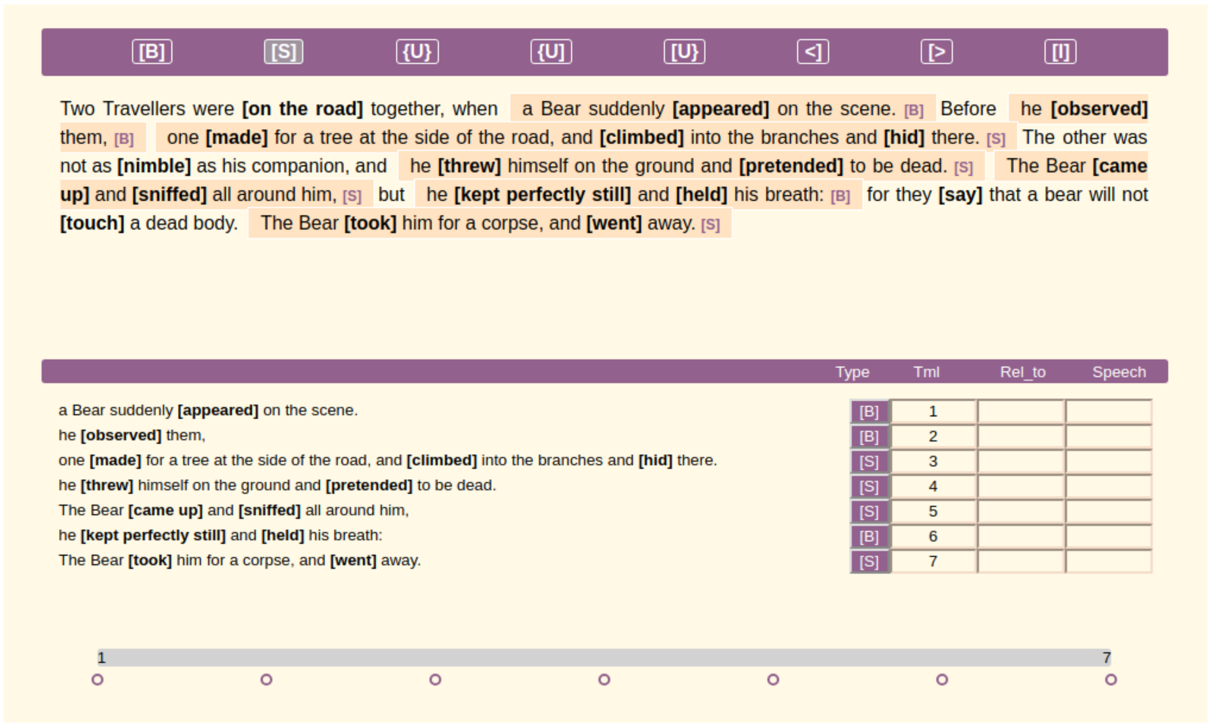
Two Travellers were **[on the road]** together, when a Bear suddenly **[appeared]** on the scene. [B] Before he **[observed]** them, one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there. The other was not as **[nimble]** as his companion, and he **[threw]** himself on the ground and **[pretended]** to be dead. The Bear **[came up]** and **[sniffed]** all around him, but he **[kept perfectly still]** and **[held]** his breath: for they **[say]** that a bear will not **[touch]** a dead body. The Bear **[took]** him for a corpse, and **[went]** away.

a Bear suddenly **[appeared]** on the scene.

Figure 4.5: Activating the temporal position input for a timeline element with a double click.

if there are spans occurring at position 1 and position 2, creating the highlights in that order will also auto-populate the Tml field with numbers “1” and “2”.

In cases where reordering is necessary, it is easy to perform by editing the corresponding numbers. **??** shows the bear story, with all the spans for [B] and [C] events highlighted in the order in which they appeared in the text.



Two Travellers were **[on the road]** together, when a Bear suddenly **[appeared]** on the scene. [B] Before he **[observed]** them, [B] one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there. [S] The other was not as **[nimble]** as his companion, and he **[threw]** himself on the ground and **[pretended]** to be dead. [S] The Bear **[came up]** and **[sniffed]** all around him, [S] but he **[kept perfectly still]** and **[held]** his breath: [B] for they **[say]** that a bear will not **[touch]** a dead body. The Bear **[took]** him for a corpse, and **[went]** away. [S]

	Type	Tml	Rel_to	Speech
a Bear suddenly **[appeared]** on the scene.	[B]	1		
he **[observed]** them,	[B]	2		
one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there.	[S]	3		
he **[threw]** himself on the ground and **[pretended]** to be dead.	[S]	4		
The Bear **[came up]** and **[sniffed]** all around him,	[S]	5		
he **[kept perfectly still]** and **[held]** his breath:	[B]	6		
The Bear **[took]** him for a corpse, and **[went]** away.	[S]	7		

Figure 4.6: The bear story with all bounded events selected in the order in which they occur.

Note how even in this short narrative the **[C]** shortcut is useful in defining several successions of events that physically could not happen in any other order: running to the tree → climbing → hiding, dropping → pretending, coming up → sniffing, being fooled → walking away. The first two **[B]** spans for single events that are steps in the narrative, and the last-but-one **[B]** span contains two processes that are bound by the events of the bear approaching and leaving.

However, the actual temporal order of the narrative is currently wrong:

- Traveller 1 was climbing the tree *while* the other threw himself on the ground,
- the bear observed them only *after* these preparations were over - or else the story would have had a very different ending.
- The bear was sniffing around Traveller 2 *while* the latter kept still.

These changes can be made by adjusting the numbers in the right column of the span table. You can use decimal rather than integer numbers to insert an event between any other already numbered events, so as to avoid massive re-numbering. For example, let's change event 2 (the bear observing the travellers) to 4.5 (i.e. before 5, which is the Bear's coming up close).

To indicate that two events were simultaneous, let us change 5 to 6, (or 6 to 5), and 3 to 4 (or 4 to 3).

The exact numbers do not matter, it is only important that their relations are correct.

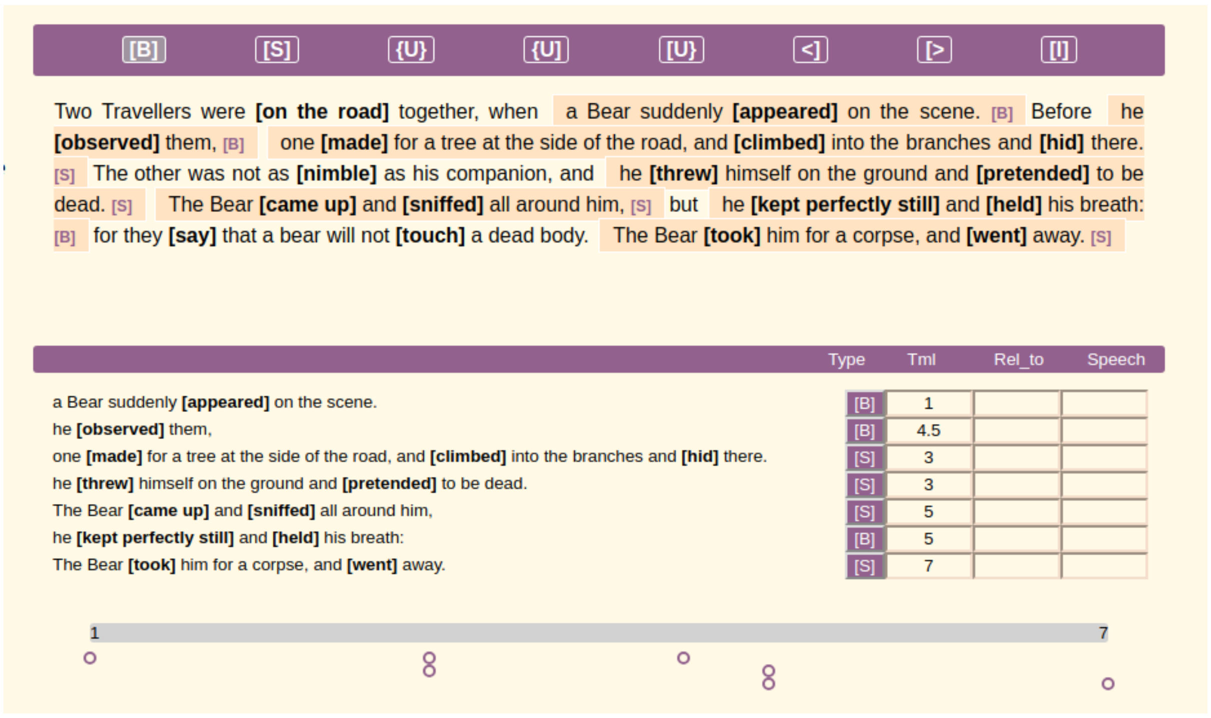
After refreshing the timeline, it looks as shown in **??**.

Once again, you can use any numbers, as long as they result in the correct temporal order. For example, if you find you need to insert an event between existing blocks that span (3:4) and (4:5), you can modify (3:4) as (3:3.5), and insert the odd block as (3.5:4).

You can see that as a result of reordering some slots on the timeline are left empty. That is not a problem, you don't need to spend time reordering everything.

4.1.3 Annotating unbounded events

Let us now look at the events for which the start/end points are not as clear. The first sentence describes the state of the two travellers throughout the story, and likely for some time before/after. In



Two Travellers were **[on the road]** together, when a Bear suddenly **[appeared]** on the scene. [B] Before he **[observed]** them, [B] one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there. [S] The other was not as **[nimble]** as his companion, and he **[threw]** himself on the ground and **[pretended]** to be dead. [S] The Bear **[came up]** and **[sniffed]** all around him, [S] but he **[kept perfectly still]** and **[held]** his breath: [B] for they **[say]** that a bear will not **[touch]** a dead body. The Bear **[took]** him for a corpse, and **[went]** away. [S]

	Type	Tml	Rel_to	Speech
a Bear suddenly **[appeared]** on the scene.	[B]	1		
he **[observed]** them,	[B]	4.5		
one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there.	[S]	3		
he **[threw]** himself on the ground and **[pretended]** to be dead.	[S]	3		
The Bear **[came up]** and **[sniffed]** all around him,	[S]	5		
he **[kept perfectly still]** and **[held]** his breath:	[B]	5		
The Bear **[took]** him for a corpse, and **[went]** away.	[S]	7		

Figure 4.7: The bear story with annotated order of all bounded events.

the NarrativeTime interface, this can be modeled as an event that spans the whole story, i.e. starting at 1 and ending at the last event (i.e. 7 in this case). In general, if you set an span to, say, 5, or 5:6, the numbers are interpreted as either the "center" or the start/end points of the state/process, and it is understood to go on for some unspecified time before/after that.

However, these numbers are redundant in cases of relatively permanent or generic characteristics. In the bear story, the Traveller 2 is likely to have been not-so-nimble for considerable time, and will probably not become an acrobat after the story, so we can treat this as a "permanent" unbounded event (section 3.3). The NarrativeTime shorcut for indicating states/processes that are generic or universal or relatively constant for the given character is setting them to ":".

After annotating these events, **??** shows that there are now two events (in green) that span the whole narrative.

4.2 Annotating discontinuous spans

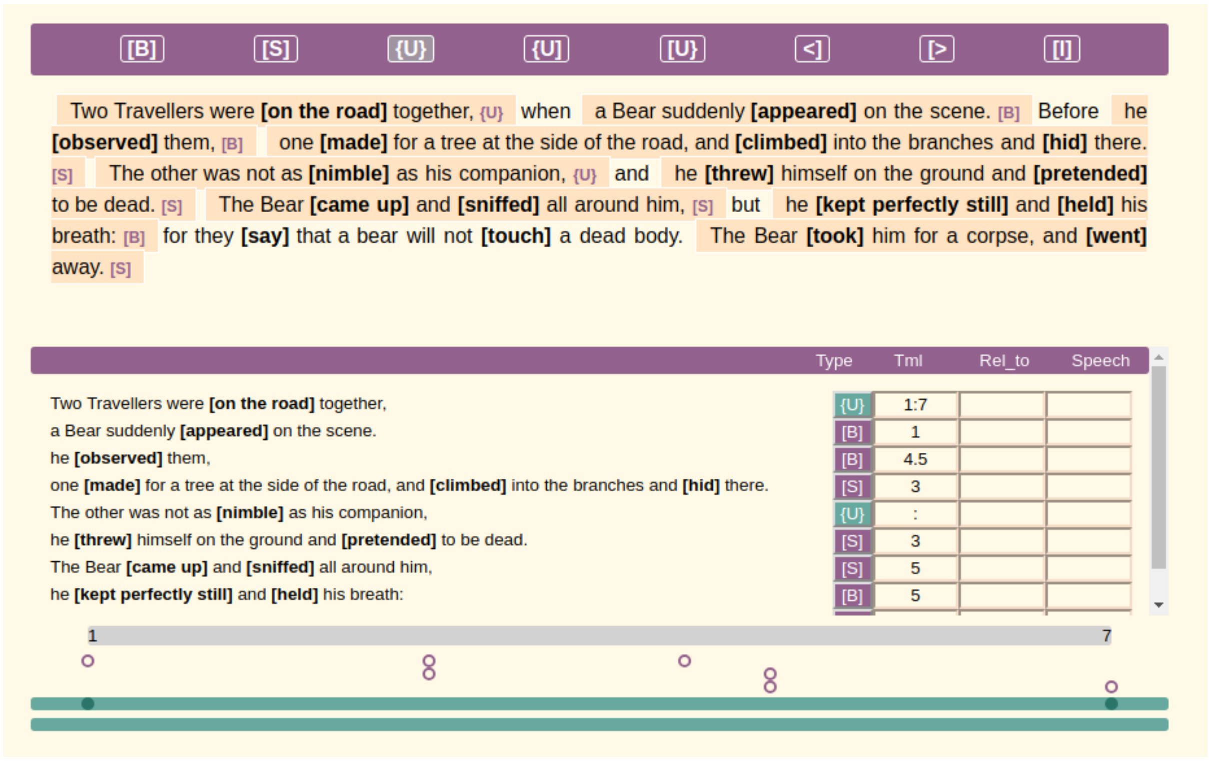
[Gutting the discontinous span annotation per se: we don't even need it if the goal is to just export to timeml. The mechanism is now only used for event coreference –ARO]

If NarrativeTime is used strictly to produce event order annotations in TimeML standard, all we need is relations between individual events. However, since it is fundamentally span-based, it is in principle possible to also use the annotations in NarrativeTime formar to extract the spans highlighted by the annotators rather than individual events (e.g. to study their chunking strategies). In that case, it may be useful to enable them to annotate discontinuous spans. Another possible use of a mechanism for linking discontinuous spans is to establish certain relations between spans, e.g. event coreference or their belonging to some type of event, a storyline etc.

NarrativeTime does have a hacky way to do this, should it be needed. Consider ex. 24:

(24) [John]$_B$, [[to [**give**] him credit]$_I$, [[**recovered**] quickly]$_B$.

The mechanism for linking discontinuous spans in NarrativeTime annotation interface is simply to append the same special symbol (!, @, #, $, %, &, ̂, *, ~) to the temporal positions of the spans that need to be linked (provided that they are of the same type and at the same temporal position). For example, let's say that John's recovering is a **[B]** event at the temporal position 2. We can link the two parts of the clause with the symbol "*" as follows:



Two Travellers were **[on the road]** together, {U} when a Bear suddenly **[appeared]** on the scene. [B] Before he **[observed]** them, [B] one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there. [S] The other was not as **[nimble]** as his companion, {U} and he **[threw]** himself on the ground and **[pretended]** to be dead. [S] The Bear **[came up]** and **[sniffed]** all around him, [S] but he **[kept perfectly still]** and **[held]** his breath: [B] for they **[say]** that a bear will not **[touch]** a dead body. The Bear **[took]** him for a corpse, and **[went]** away. [S]

	Type	Tml	Rel_to	Speech
Two Travellers were **[on the road]** together,	{U}	1:7		
a Bear suddenly **[appeared]** on the scene.	[B]	1		
he **[observed]** them,	[B]	4.5		
one **[made]** for a tree at the side of the road, and **[climbed]** into the branches and **[hid]** there.	[S]	3		
The other was not as **[nimble]** as his companion,	{U}	:		
he **[threw]** himself on the ground and **[pretended]** to be dead.	[S]	3		
The Bear **[came up]** and **[sniffed]** all around him,	[S]	5		
he **[kept perfectly still]** and **[held]** his breath:	[B]	5		

Figure 4.8: Annotating unbounded events.

Event		Tml	Rel_to	Speech
John,	**[B]**	2*		
to [**give**] him credit,	**[I]**	:		
[**recovered**] quickly	**[B]**	2*		

If you have another discontinuous span that is also at the position 2, simply use a different symbol. The current version of the export script treats the combination of the same temporal position and the same special symbol as spans referring to the same event(s); if you would like to use NarrativeTime for span annotation, you would only need an export script to reflect the way you intend to use the discontinuous spans.

5 Branching timelines

5.1 What are branching timelines and where they attach

So far we have been assuming a simple narrative with a single timeline, in which all the events can be unambiguously placed. In such narratives we can tell what the temporal relation is between any two events on the timeline, no matter how far away they are.

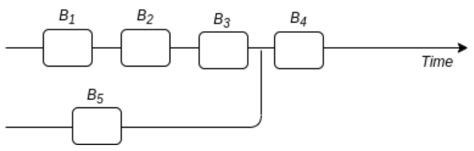
In the actual texts, this is often not the case: we may not always know the ordering of all events with respect to all other events. However, in a coherent story it is not feasible that an event has no temporal relation to *any* other event. What does happen is partially specified or relative order: an event may have a known temporal relation to *some* other events, but not all of them.

To handle this, NarrativeTime introduces the mechanism of branching timelines. Consider ex. 25:

(25) [Mary [**studied**] biology in high school]$_B$^{1}. [She [**won**] a scholarship]$_B$^{2} and [[**went**] to college]$_B$^{3}. <...> [John had [**served**] in the army]$_B$^{4} before [he [**married**] Mary]$_B$^{5}.

The events B_1-B_3 are consecutive, and B_5 presumably follows them. But what about B_4? We don't know if John joined the army while Mary was applying for the scholarship, or while she was already in college, or before that. All we know is that it happened before the marriage event. We can indicate precisely that using the branching mechanism, which creates a partial parallel timeline attaching to the main timeline at the marriage event. This indicates that B_4 happened before B_5, but its relation to B_1-B_3 is not specified.

Here is how we could represent it schematically:



Note that since the first (or only) events on branching timelines are the first/last on their respective timelines, they enjoy the same privilege as the first/last events in the overall narrative: the initial temporal boundary for the past events, or the final boundary for the future events. They are always either fully or partially bounded.

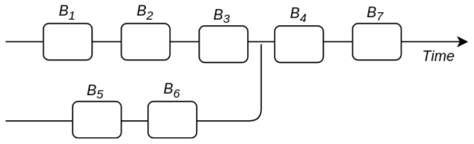
Let us consider another example:

(26) [John [**returned**] to New York]$_B$^{1}. <...> [John [**bought**] the ticket]$_B$^{2}, [had a quick [**coffee**]]$_B$^{3} and [[**headed**] to the movie theater]$_B$^{4}. [He had already [**read**] the book]$_B$^{5} and [he [**liked**] it]$_B$^{6}. [The movie [**started**]]$_B$^{7}.

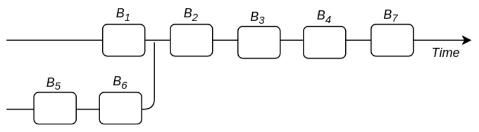
Let us assume that it is not clear from the text whether John read the book before or after coming to New York. However, it is clear that he did that before watching the movie. In normal reading you would not stop and try to figure out the timeline of the reading event vs returning to New York, and in NT annotation we should not have to worry about that either. We can simply put the reading event on an alternative timeline that joins the main timeline before the movie starts.

The exact position at which the branching timeline joins the main one may be contentious, as this is not specified in the text. Generally there are two strategies for this:

- **Attaching the branch to the nearest event at which it is mentioned**, i.e. the closest event that precedes in the text the events in the branch. In this case it is John's heading to the theater:



- **Attaching the branch to the point in the narrative where it makes the most sense according to our knowledge of the world**. We do not know if John read the book before or after or while coming to NY, but he probably did so before deciding to go to see the movie and buying the movie ticket, because these are both subevents of the visiting-the-movie-theater schema. Thus we could represent the sequence as follows:



In practice, we find that both strategies are hard to apply consistently. In some case the natural text interpretation flow strongly suggests looking for the anchoring point that makes the most sense according to the annotator's world knowledge, e.g. because the individual annotator has a strong association or personal experience making a certain interpretation more salient. But in many cases the same annotator simply would not have an equally strong preference for one timeline interpretation over another. For example, if a text mentions a meeting with a prospective husband and the wedding, and in a different section there is a mention of their first kiss, it is hard to not mentally place the kiss somewhere between the meeting and the wedding - but we would not be able to easily tell if it happened after the first or second date. And we should not even try.

Suggested rule of thumb: the branches should attach to the main timeline at the point at which their events are mentioned, unless you have a strong preference to attach them at some earlier (for past events) or later (for future events) point. If you find yourself overthinking any attachment point, just let it be where it is mentioned.

5.2 Creating branching timelines in the annotation interface

Figure 5.1 shows how branching timelines are represented in the annotation interface. For each branch, a triangle element appears on the main timeline (the shape corresponding to the < > symbols which, as described above, indicate events happening before or after the attachment point). Hovering over the triangle brings up a mini-timeline, the elements of which can be inspected in the same way as for the main timeline.

Note that in this case the annotation in Figure 5.1 is done in the way that minimizes the annotation required: three [**B**] events on the main timeline are combined into a single [**C**]. When a before-branch ("<") is attached to [**C**] sequence, it is interpreted as "the first event in the sequence is the anchor". Accordingly, anchoring an after-branch (">") to a [**C**] sequence, is interpreted as "the last event in the sequence is the anchor".

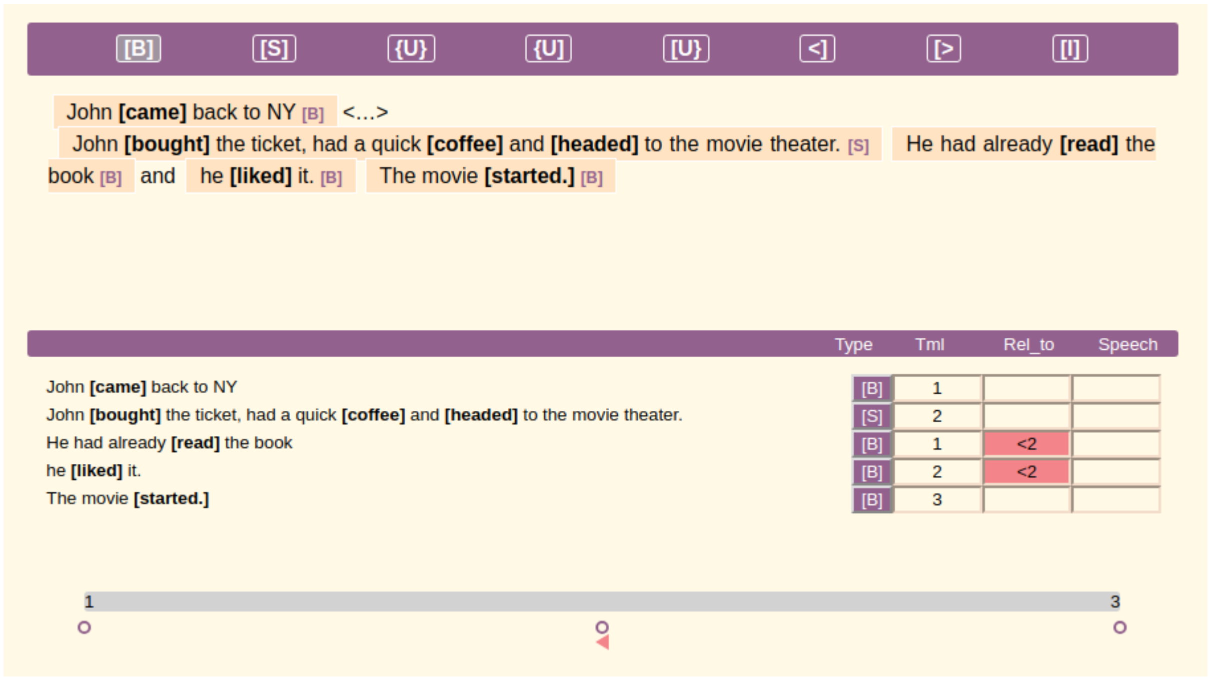
Generally there are two ways to create branching timelines: manually and with the shortcut buttons [>] and <].

Method 1: manual input for the already-defined events. The events on branching timelines are the regular bounded/unbounded/partially-bounded events that you can create with all the regular buttons like [**B**] and [**C**]. If you have created some events on the main timeline, and find that they should be on a branching rather than the main timeline, you can move them to a branch by editing the Rel_to input column in the annotation table. It indicates with respect to what event the current sub-timeline is defined.

Consider a simple example:

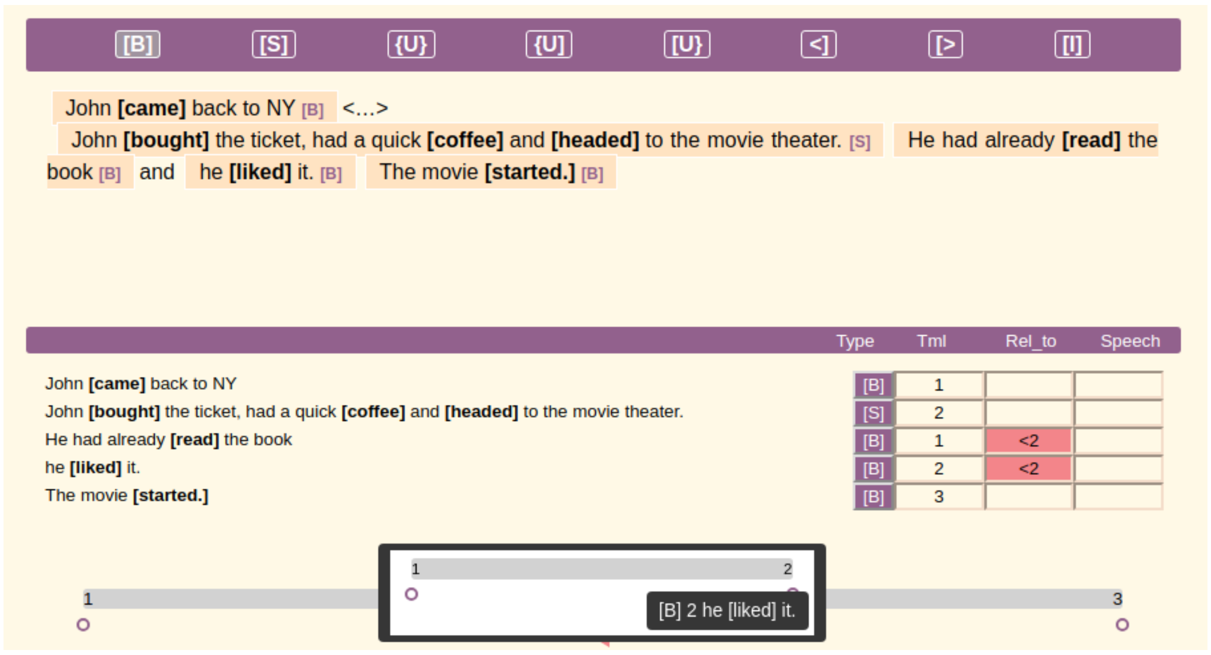
(27) <some events from John's life> [John [**headed**] to the theater]$_{\text{B}}$. [He had [**read**] the book already]$_{\text{B}}$.

Let us assume that there are multiple events from John's life that we cannot position on the main timeline with regards to his reading the book, which would justify putting that event on a branching timeline. Let us further assume that he heads to the theater at the temporal position 20. His reading the book is a [**B**] event that can be represented as follows:



John **[came]** back to NY [B] <...>
John **[bought]** the ticket, had a quick **[coffee]** and **[headed]** to the movie theater. [S] He had already **[read]** the book [B] and he **[liked]** it. [B] The movie **[started.]** [B]

	Type	Tml	Rel_to	Speech
John **[came]** back to NY	[B]	1		
John **[bought]** the ticket, had a quick **[coffee]** and **[headed]** to the movie theater.	[S]	2		
He had already **[read]** the book	[B]	1	<2	
he **[liked]** it.	[B]	2	<2	
The movie **[started.]**	[B]	3		



John **[came]** back to NY [B] <...>
John **[bought]** the ticket, had a quick **[coffee]** and **[headed]** to the movie theater. [S] He had already **[read]** the book [B] and he **[liked]** it. [B] The movie **[started.]** [B]

	Type	Tml	Rel_to	Speech
John **[came]** back to NY	[B]	1		
John **[bought]** the ticket, had a quick **[coffee]** and **[headed]** to the movie theater.	[S]	2		
He had already **[read]** the book	[B]	1	<2	
he **[liked]** it.	[B]	2	<2	
The movie **[started.]**	[B]	3		

[B] 2 he [liked] it.

Figure 5.1: Branching timelines in the annotation interface

Event		Tml	Rel_to	Factuality
John [**headed**] to the theater	[**B**]	20		
He had [**read**] the book already	[**B**]	1	<20	

This notation means that in the sub-story leading up to John's going to the theater, the first event is that he had read the book, and this event does not have preceding/following relations with any events prior to 20 on the main timeline (i.e. these relations are underspecified, the reading could happen before event at position 5, 10, 15, or any other less than 20).

For events that happened BEFORE a given event, the notation is "<" + the timeline position of that event, e.g. "<20". Similarly, "20>" indicates a branching timeline that happens AFTER a given event.

Method 2: Shortcut buttons. In the annotation interface, the Rel_to field gets auto-populated if you use the <] or [>] buttons in the annotation interface to select the span. Creating a span with this type selected will perform three actions automatically:

1. create a [**B**] event (as it is the most common case);
2. auto-fill the Rel_to column with the latest known timeline position number (20 in this case), and add "<" to indicate that the branching timeline events happen *before* that point;
3. auto-fill the Tml column with position 1, to indicate that in the branching timeline this is the first event.

If your branching timeline has more events, for as long as <] is selected, any subsequent spans will be created as [**B**] events at position 2, 3, 4 etc. in the same branching timeline. If you need to change the span type of any of them to [**C**], **{U}** etc., this can be done manually in the annotation table by clicking the span type button.

5.3 Factuality

So far we have considered events for which it is clear from the context that they did in fact happen. But what if that is unclear, if the event is in the future or a possibility? What about negated events - did they happen, should they be on the timeline at all? If so, what event type should they be?

NT solution is to treat them exactly like regular events, but also mark their factuality status. For example, a future can still be an event of type **[B]** or **{U}**, preceding or following other events, on the main or a branching timeline. If it is not certain that it actually happens, we can just mark it as a "maybe will happen" event. And if the event is negated, we can simply mark that too.

Marker	Example Sentence	Factuality
[**event**]	[John [**ate**] a cake]$_B$.	Happened / will happen
[**event**]$_-$	[John didn't [**eat**]$_-$ a cake.]$_B$	Did not happen
[**event**]$_{M+}$	[John may have [**eaten**]$_{M+}$ a cake.]$_B$	Maybe happened / will happen
[**event**]$_{M-}$	[John probably didn't [**eat**]$_{M-}$ a cake.]$_B$	Maybe did not / will not happen

Table 5.1: Event factuality in NarrativeTime

5.3.1 "Happened/will happen" events [unmarked]

These are the events for which a reader of the story does not doubt their factuality. Most of marked events in narrative texts should fall into this category, both in fiction and non-fiction. Since this is the majority, default category, they do NOT need to be marked in the annotation interface. We estimate that for a typical non-fiction narrative less than 20% of events would fall in one of the other categories.

(28) [John [**watched**] football with Fred on Saturday.]$_B$

5.3.2 "Maybe happened/will happen" events [M+]

These are the events that might have happened or could happen, but we are unsure about it. For instance, in example 29 (a) we may believe that Mary will call John (rather than that she will not) but we cannot tell for sure as this is a future event. Similarly, in 29 (b) while we don't know whether John met Mary we are likely to believe, as does the speaker, that he did.

(29) a. [Mary promised to [**call**]$_{M+}$ John on Monday]$_B$.
b. [John must have already [**met**]$_{M+}$ Mary.]$_B$

Note: while factuality of future events is normally not 100% guaranteed, sometimes we do accept it as a given. One example is the "narrated future": a future event that is communicated by some kind of omniscient narrator.

(30) John met Mary. Little did he know that [the next day they would be [**married**]]$_B$[2].

5.3.3 "Didn't happen" events [-]

By definition, negatives are the events that did not happen, so they would have 0 on the factuality dimension. However, the fact of something not happening may itself be informative, a meaningful event (for example, consider *not* calling your mother on her birthday). Furthermore, such non-events have a very precise position on the timeline, the same as any other event. So NarrativeTime policy for such events is to determine their type and position on the timeline in the same way as if they were positive events. 31 (b) and 32 (b) are both examples of such events.

(31) a. [Mary [**called**] John on Monday]$_B$.
b. [Mary did not [**call**]$_-$ John on Monday]$_B$.

(32) a. {Mary was [**sick**]}$_U$ when I called her.
b. {Mary was not really [**sick**]$_-$}$_U$ when I called her.

5.3.4 "Maybe didn't happen/won't happen" events [M-].

These are events that may not have happened, or may not occur in the future: the negated events for which we are unsure about their factuality status. For instance, in ex. 33 (a) we don't know whether Mary will come or not, but we know that it is possible that she will not. Similarly, in ex. 33 (b), we do not know whether John pass the exam, but we know that he could have failed. All negated possibilities will fall into this category.

(33) a. {Mary [**may**] not [**come**]$_{\mathrm{M-}}$ today}$_{\mathrm{U}}$.
b. {It's [**possible**] that John didn't [**pass**]$_{\mathrm{M-}}$ the exam}$_{\mathrm{U}}$.

In the NT annotation interface, these four categories are expressed as additional annotations in the annotation table:
[TODO screenshot –ARO]

5.3.5 Difficult cases in annotating factuality

To reiterate, NarrativeTime focuses on annotating the *order* of events. The annotation tool presupposes that the events themselves are already marked, and they can be marked according to any annotation scheme developed for that purpose. That scheme may focus on events expressed by verbs, strictly-factual events, or anything else. But since there is a lot of variation in what is considered an event, Table 5.2 and Table 5.3 present our take on some difficult types of events and proposed approaches to annotating their order.

A particularly tricky choice for any annotation scheme is the complex events, such as the expressions of modality and intentions. Any annotation scheme for events has to choose whether they are treated as a single event, two separate events, or only one of them is annotated. Consider ex. (34) and (34):

(34) a. John [**wants to study**] Spanish.
b. John [**wants**] to [**study**] Spanish.
c. John [**wants**] to study Spanish.
d. John wants to [**study**] Spanish.

(35) a. John [**can swim**] a mile.
b. John [**can**] [**swim**] a mile.
c. John [**can**] swim a mile.
d. John can [**swim**] a mile.

If a given project chooses the option (c) or (d) for such events, Table 5.3 presents our suggested solutions. Option (b) is in fact a combination of (c) and (d), so both solutions we propose can be used. Option (a) needs an explicit decision of what is the being annotated: the modal/intention or the possible event. That decision then breaks down to either option (c) or (d).

Event type	**Rationale & Examples**	**Factuality**
Hypotheticals: conditions of other events	"Maybe/maybe not" for conditions in present/future, "not happened" for the past • If this is [**true**]$_{M+}$, he will go to school tomorrow. • If this were [**true**]$_{M+}$, he would come to school. • If this had been [**true**]$_{-}$, he would have come to school.	+,-, M+, M-
Hypotheticals: consequences of conditions	"Maybe/maybe not" for consequences in present/future, "not happened" for the past • If this is true, he will [**go**]$_{M+}$ to school tomorrow. • If this were true, he would have [**come**]$_{M+}$ to school. • If this had been true, he would have [**come**]$_{-}$ to school.	+,-, M+, M-
Imperatives, calls to action, advice	The course of action in the request/advice may not be followed • [**Vote**]$_{M+}$ for your future! • Don't [**talk**]$_{M-}$, please. • When you see John, [**tell**]$_{M+}$ him to call me.	M+, M-
Questions	By definition, if an event is the focus of a question, its factuality is unknown: • Who is [**listening**]$_{M+}$? At the same time, questions may contain events presented as given information. Those are annotated as usual: • I have [**stated**] my views in press on several occasions, but who listens to scientists nowadays?	M+, M-
Future events	Future is uncertain, so these events may not happen. • Mary will not [**call**]$_{M-}$ John. • When you [**see**]$_{M+}$ John, tell him to call me.	M+, M-
Potentially unreliable sources	If we suspect that the narrator is not fully reliable (deliberately or not), their information may not be accurate. • According to John, Marry didn't [**get married**]$_{M-}$ last week. But this depends on the context. If we trust John, the same events should be annotated as factual. And note that the communication event itself is factual, if it's explicit. • John [**said**] that Marry got [**married**]$_{M+}$ last week.	M+, M-
Hedging: opinion, belief, perception	In dialogue, the speaker may emphasize that X is their own opinion to express uncertainty. In a narrative the belief states of characters may also not be be perceived as reliable sources of information. If you as a reader doubt that the event actually happened, the "maybe" factuality is appropriate. The belief events themselves are factual. • I [**believe**] that Marry was [**wrong**]$_{M+}$. • John [**thought**] that Marry was [**wrong**]$_{M+}$. • The situation [**seemed**] [**surreal**]$_{M+}$ to John.	M+, M-
Metaphors, idioms, figures of speech	Figurative language is intended as an accurate description of the true state of events • John is a [**night owl**]. • John is [**snowed under**] with work.	+,-

Table 5.2: Annotating future & hypothetical events, imperatives, questions, unreliable information

Event type	**Rationale & Examples**	**Factuality**
Possibility, need, obligation, permission etc.: the expressions thereof	The fact that X is possible/needed etc. is a real, factual state (most often a temporary state) • Mary [**could**] swim all day. • Mary [**needed**] to swim every day. • Mary [**had to**] swim every day. • Mary was perfectly [**capable**] of swimming.	+,-
Possibility, need, obligation, permission etc.: the events that are possible/needed/permitted etc.	These events (bounded or unbounded, depending on the context) may or may not happen. • Mary could [**swim**]$_{M+}$ all day. • Mary needed to [**swim**]$_{M+}$ every day. • Mary had to [**swim**]$_{M+}$ every day. • Mary was perfectly capable of [**swimming**]$_{M+}$.	M+, M-
Intentions, wishes, goals, attempts, preference: the expressions thereof	The event of intending/wishing/trying is a real, factual event (usually bounded) • Mary [**promised**] to call John on Monday. • John [**went**] to NY to meet Fred. • John didn't [**want**]$_{-}$ to learn Spanish. • John [**tried**] to learn Spanish.	+,-
Intentions, wishes, goals, attempts, preference: the events that are intended/promised etc.	An event that is intended/promised/wished for is only a possibility • Mary promised to [**call**]$_{M+}$ John on Monday. • John went to NY to [**meet**]$_{M+}$ Fred. • John didn't want to [**learn**]$_{M+}$ Spanish. • John tried to [**learn**]$_{M+}$ Spanish.	M+, M-
Comparatives: the actual events	Comparison is typically drawn for events that do happen: • John was [**swearing**] like he used to in the old days. • John was [**swearing**] as if he was Irish. • This plan is as [**cunning**] as a fox who's just been appointed Professor of Cunning at Oxford University.	+,-
Comparatives: the compared-to events	The compared-to events can be either real or not: • John was swearing like he [**used to**] in the old days. • John was swearing as if he was [**Irish**]$_{-}$. • This plan is as cunning as a fox who's just been [**appointed**]$_{-}$ Professor of Cunning at Oxford University.	M+, M-, +,-

Table 5.3: Annotating factuality: modals, intentions, comparatives

6 Anchoring temporal expressions to events

Temporal expressions ("timex" in TimeML terminology) are a frequent feature of temporal annotation schemas. It may be an absolute indicator of time (e.g. date, time of day), a relative indicator (e.g. "two hours earlier"), an expression of duration (e.g. "for two hours") etc. Similarly to events, we assume that the timexes are already annotated according to any schema you may choose.

The goal of NarrativeTime is to establish links between them and the events on the timeline. The span-based approach of NarrativeTime enables performing such annotation simultaneously with event ordering, simply by including the timexes in the same spans as the events they should be related to. For example:

(36) a. [John [**came**] on Friday]$_B$.

b. [John was [**born**] in 1980]$_B$.

c. [John was [**singing**] from 9 to 5]$_B$.

If there are several consecutive **[B]** events and one of them is associated with a temporal expression, you should not use the **[C]** shortcut for indicating the sequence, because it will be impossible to tell which event the temporal expression should be linked to. For example:

(37) [John [**went**] home, [**took**] a shower]$_C$, and [[**went**] to bed at 10 pm]$_B$.

The last span is separated off because "10 pm" only applies to going to bed, not to showering and heading home. Annotating it as one chain would mean that all three events happened at 10 pm.

[TODO: add a screenshot –ARO]

7 Tips for working with the annotation interface

7.1 Keyboard shortcuts

B	activating [B] span creation mode
B	activating [C] span creation mode
U	cycling through **{U}**, [U}, **{U**] span creation modes
R	cycling through <], [>] span creation modes

Since we define **{U}**, **{U**] and [U**}** events as we read the text, we do not yet have the complete knowledge of the story. This means that what is defined as [U**}** may actually end up being [**B**], because at some later point the writer may specify the end of that process explicitly. For instance, at timeline position 15 we may have been told that John got fat at some point in the story, and we mark it as [U**}**15:. But then at position 25 we are told that he lost weight again.

In such cases the event of John being fat is actually the same, only its temporal specification that changed. To mark that, we need to go back to the original **{U**] at location "15:" and change it to [**B**] at location "15:25". Fortunately, that does not happen often, and when it does the narrator explicitly draws our attention to it (e.g. "something was over", "something ended", etc.).

7.2 Interface actions

Importing an annotation task: Use the "Import file" button on the left-side control panel, and select the text to annotate with the drop-down menu or arrows.

Saving your work: "Save" button saves intermediate results during one work session; they will be lost if you reload your browser or switch annotation mode. "Export" on the left-side control panel saves your work to a file on the hard disk.

Deleting an annotation: click on the corresponding span in the text.

Changing the type of an event span after it was created: click on the event type button in the annotation table.

Checking the overall timeline: hover over elements on the timeline to see the associated text. Clicking on an element activates the corresponding Tml input field.

Starting over: Click the "Clear all" button in the left-side control panel. TODO: CURRENTLY DOES NOT RESTART THE COUNTER

Annotating discontinuous spans: append a shift-character to the timeline position of both parts of a discontinuous span (assuming it is of the same type and in the same position). Supported characters: !, @, #, $, %, *,ˆ, !, ?, +. See section 4.2

I've annotated most of the text, and now I noticed something that should have come before/in the middle of the existing timeline! The actual numbers in the Tml column do not matter, it is only important that the order is correct. Any number that is not already used can be used for creating new timeline elements: -10, 3.95, 100, whatever.

7.3 How to annotate in the narrative order

The fastest way to annotate is to avoid having to specify the numbers for the timeline positions - by simply highlighting the spans in the right order! This way you create two pieces of information in one action: defining the span itself and its position on the timeline.

This can be achieved as follows.

1. Read the full text to understand the general storyline.
2. Check if it is all in the narrative order, or there is some kind of temporal mixture (e.g. the narrator is talking about the past but also mentioning the present sometimes).
3. Go through the story, highlighting the spans of [**B**] and [**C**] spans as appropriate. You only need to break [**C**] spans if there are timexes (chapter 6) or change of factuality status (section 5.3).
4. If there are branching timelines, you can trigger the special <] and [> input mode to autofill both Tml and Rel_to columns (chapter 5). Note that the branch will be automatically attached to the latest position of the [**B**] or [**C**] type, so if you first create the anchor event input will be simplified. De-activating this mode will let you continue with the auto-input of [**B**] and [**C**] timeline positions as before.
5. Any unbounded events that are anchored in the [**B**] and [**C**] event positions can be annotated by simply activating the corresponding input mode with the "u" shortcut and creating the span *while the anchor position is the last on the timeline.* This will automatically insert that position in the Tml column, so frequent cases like "John saw Mary, she was wearing a red dress." are easy to handle. You will only need to edit the Tml field for "permanent" **{U}** events (position ":") or when the anchor event is at a different position.

If there are two or more major temporal "positions" in the story (e.g. the narrator tells about something in the past while also making remarks in the present tense), to make use of auto-numbering you may want to work through the "past" parts first.

It also helps to keep the distinct clusters visually distinct on the storyline by selecting the starting points that are sufficiently remote, e.g. the past events start at position "1" on the timeline, and the present starts at "30". This will allow you to gradually fill in the past storyline without counting how many narrative steps you actually need there.

Check the timeline often! Hovering over any point or span on the timeline brings up the text of the span, which should be helpful for correcting the numbered order of events.

7.4 Cheatsheet

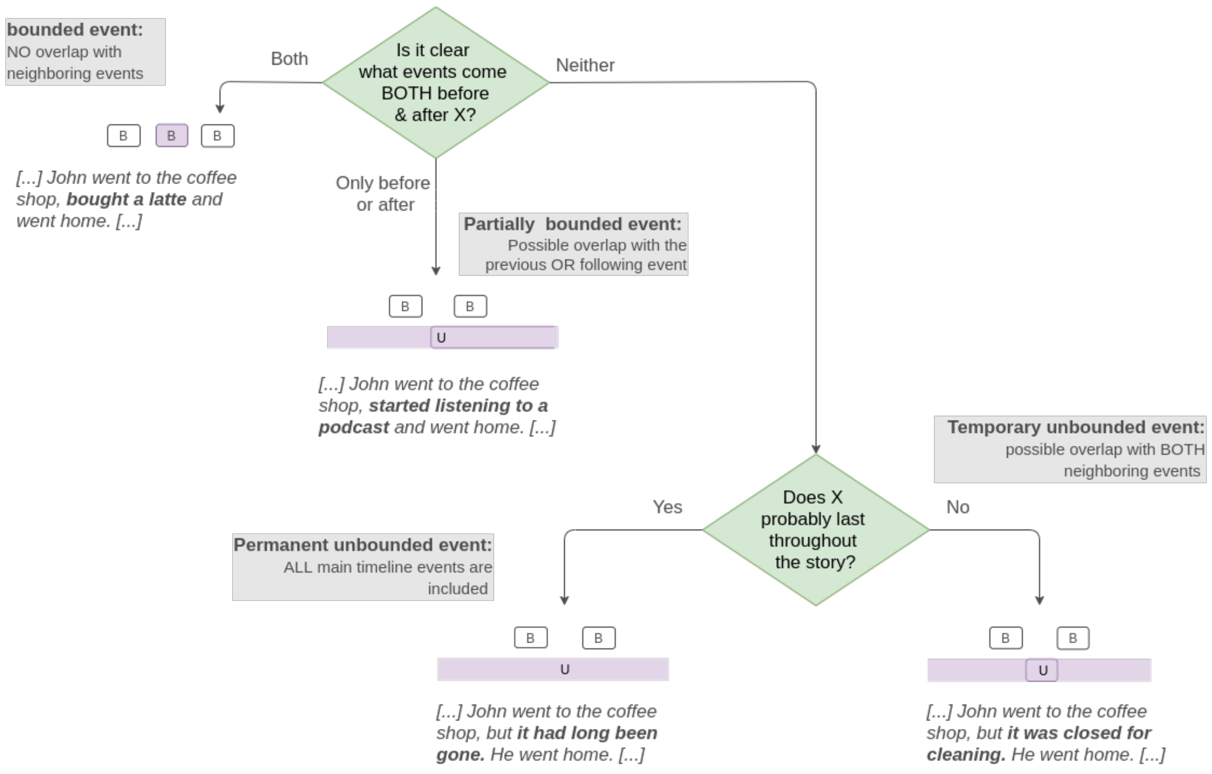

bounded event:
NO overlap with neighboring events
Both
Is it clear what events come BOTH before & after X?
Neither
B
B
B
[...] John went to the coffee shop, bought a latte and went home. [...]
Only before or after
Partially bounded event:
Possible overlap with the previous OR following event
B
B
U
[...] John went to the coffee shop, started listening to a podcast and went home. [...]
Temporary unbounded event:
possible overlap with BOTH neighboring events
Yes
Does X probably last throughout the story?
No
Permanent unbounded event:
ALL main timeline events are included
B
B
U
B
B
U
[...] John went to the coffee shop, but it had long been gone. He went home. [...]
[...] John went to the coffee shop, but it was closed for cleaning. He went home. [...]

Figure 7.1: What type of event is it?

Since to determine the type of event we need to know the preceding and following events, the first and the last events in the story are tricky: we simply are not told what comes before/after them. **The NarrativeTime convention is to always treat the beginning and the ending of the text as a boundary**. The first/last events can thus be either bounded or partially bounded (see section 3.4).

How to save effort?

- clusters of roughly-simultaneous events: together in the same [**B**] or the "temporary" **{U}** spans
- clusters of generic events/permanent characteristics: together in the same "permanent" **{U}** span
- consecutive bounded events: together in a single [**C**] span

Clusters of events can be annotated together **as long as the events in the cluster have the same factuality status**. For [**C**], there should also be no temporal expressions, other events that are simultaneous with a part of the sequence, or branches from events in the sequence.