Proceedings of ISA-18 Workshop at LREC2022, pages 33–41
Marseille, 20 June 2022

Event Sequencing Annotation with TIE-ML

Damir Cavar, Ali Aljubailan, Ludovic Mompelat, Yuna Won, Billy Dickson, Matthew Fort, Andrew Davis, Soyoung Kim
Indiana University
Bloomington, USA
{dcavar, aaaljuba, lmompela, yunawon, dicksonb, mattfort, ad7, sk135}@iu.edu

Abstract

TIE-ML (Temporal Information Event Markup Language) first proposed by Cavar et al. (2021) provides a radically simplified temporal annotation schema for event sequencing and clause level temporal properties even in complex sentences. TIE-ML facilitates rapid annotation of essential tense features at the clause level by labeling simple or periphrastic tense properties, as well as scope relations between clauses, and temporal interpretation at the sentence level. This paper presents the first annotation samples and empirical results. The application of the TIE-ML strategy on the sentences in the Penn Treebank (Marcus et al., 1993) and other non-English language data is discussed in detail. The motivation, insights, and future directions for TIE-ML are discussed, too. The aim is to develop a more efficient annotation strategy and a formalism for clause-level tense and aspect labeling, event sequencing, and tense scope relations that boosts the productivity of tense and event-level corpus annotation. The central goal is to facilitate the production of large data sets for machine learning and quantitative linguistic studies of intra- and cross-linguistic semantic properties of temporal and event logic.

Keywords: TIE-ML, event sequencing, semantic annotation, temporal logic

1. Introduction

Natural languages provide different means to encode properties of events and their relative order along the time axis in discourse. Tense, aspect, mood, and modality make up the foundations of this encoding, and various combinations of these tools are employed cross-linguistically. Whether these features are expressed lexically, morphologically, as prosodic properties, or whether they need to be induced from semantic or pragmatic cues during a conversational process, we assume that tense places events on a timeline and establishes relations between them, while aspect describes the flow of time respective to the event.

1.1. Cross-linguistic Variation

When comparing different languages, we can identify significant differences between expressions of events and time which we aim to capture for both quantitative studies, and for the development of machine learning tools that automatically annotate and process text.

English and other Germanic languages for example inflect the main verb for past and present tenses and use a modal auxiliary to form the future tense. Perfect aspect is expressed using periphrastic forms though the auxiliary construction *have* + *past participle* and progressive aspect is expressed through the combination of *to be* + *-ing*. To contrast, Japanese and Korean mark for past and non-past, and rely on adverbials or context to place events in the future. They can also employ an intention construction in the present tense to indicate future events, and Korean additionally allows the reduplication of the past tense suffix to place events in the remote past. In Semitic languages, for example Arabic and Hebrew, there is an ambiguous association between tense and aspect in verb conjugation.

This diversity of grammatical properties related to tense and event interpretation was one of the motivations for the corpus project discussed in this article. The other motivation was to compare the observed effects of scope relations between predicates in complex sentences (multi-clause structures) on the interpretation of tense for each individual clause. In this context we observe that there are differences between types of subordinate clauses such that some are subject to *Tense Agreement*, while others are not restricted with respect to their tense by any dominating clause. In other words, while some clauses have to agree in tense with their dominating clause, for others, the semantic tense is determined by the dominating clause, and the morphological or periphrastic tense interpretation is altered. In the following section we explain these aspects of the impact of structural scope relations on the interpretation of tense, and the need for corpora to study the qualitative and quantitative properties of those cross-linguistically.

1.2. Interactions between Syntax and Semantics

Interesting research questions related to events and tense emerge from the study of the interaction of temporal and event properties in complex sentences. Scope relations between clauses determine the interpretation of tense associated with a clause level predicate, see for example the discussion of the sequence of tense puzzle in Kiparsky (2002).

The interpretation of the past tense predicate in (1) includes the assumption that an event occurred that resulted in the fact that *Apple* and *Alphabet* are now a single organization. The past tense implies factivity or a positive truth value.

(1) *Apple merged with Alphabet.*

Placing this clause in the scope of a simple matrix clause with a past tense predicate does not change this interpretation, as example (2) shows.

(2) *Reuters reported*
[*that Apple merged with Alphabet*]

The interpretation of the embedded clause in example (2) does not change significantly in comparison to example (1). However, if we alter the tense of the matrix clause, the interpretation of the temporal properties of the embedded predicate is altered significantly, as example (3) shows.

(3) *Reuters will report*
[*that Apple merged with Alphabet*]

Example (3) no longer allows for the assumption that *Apple* indeed *merged with Alphabet* at speaker time, nor is it possible to exclude this possibility. The future tense of the matrix clause provides a new time frame that affects the past tense interpretation of the embedded clause.
We also observe that some clauses that are syntactically assumed to modify a predicate, tend to agree with respect to tense with the modified predicate. Clauses that are selected by the predicate, as with *report* and the subordinate clause in (3), do not have to agree for tense in the same way. We assume that example (5) in contrast to (4) should be considered deviant and semantically problematic, if not completely ungrammatical.

(4) *While I will be in Paris, Reuters will report*
[*that Apple merged with Alphabet*]

(5) ? [*While I was in Paris*] *Reuters will report*
[*that Apple merged with Alphabet*]

Similarly, deeper clause embedding of modifying predicates as in (6) is subject to the same kind of constraint. The ungrammaticality of (7) is due to the mismatch between the tense in the modifier headed by *visit* and the matrix clause head predicate *reported*.

(6) *During the time that I visited Paris, Reuters reported that Apple merged with Alphabet.*

(7) **During the time that I will visit Paris, Reuters reported that Apple merged with Alphabet.*

In some cases this compatibility of tense properties between modifier clause and modified predicate is even more complex, as the *if*-clauses in (8) and (9) show.

(8) *If I visit Paris, Reuters will report that...*

(9) **If I will visit Paris, Reuters will report that...*

Without going into details why present tense of the modifier predicate is compatible with the future tense of the modified predicate, we observe that morpho-syntactic present tense is compatible and often used as semantic future tense.
We observe that modifying temporal adverbial clauses (adjuncts) need to agree in tense with the head predicate that they modify. Selected predicates either have to have tense or be infinitival, but the tense of a selected finite clause is not part of the selection.
To be able to capture the clause-level tense properties and the complex interactions of tense scope induced via syntactic properties, we have to take into account multiple annotation levels or tiers, to describe for example:

- syntactic scope relations (dominance and precedence at least),
- the tense of the particular clauses, and
- the semantic relations between clauses in terms of selection vs. modification.

Thus in the annotation approach discussed here, the information about clause hierarchy, selection or modification, as well as linear sequencing have to be captured.

1.3. Sequencing and Duration of Events

As emphasized in Cavar et al. (2021), sequencing of predicates and events along the time axis is a phenomenon related to tense and the temporal interpretation of predicates. In example (10) we observe that the linear order of sub-events as presented in syntax correlates with the semantic alignment of the sub-events along the time axis.

(10) *Wash the veggies, chop them, and fry them.*
1 2 3

Example (11) on the other hand exhibits a mismatch between the linear presentation sequence of sub-events and the underlying temporal alignment on the time axis.

(11) *Before you fry the veggies, make sure to wash*
3 1
and chop them.
2

Capturing the empirical sequencing of events along the time axis, their potential coincidence, or their overlap is relevant at various levels in semantically aware Natural Language Processing (NLP) or Artificial Intelligence (AI) applications. In computational reasoning, the temporal sequence of events is often relevant for the analysis of causality or intention.
The central focus in the first phase of this effort was to provide temporal sequencing annotations for events in complex sentences. There are many other highly relevant issues related to this, as for example duration of events. By breaking the temporal annotation effort and ML model training into atomic tasks, we expect to achieve faster much better results. In the next project

phase we are extending the effort to annotation of duration and overlaps of events.
We have not identified any other resource that could provide enough cross-linguistic data sufficiently large to train ML models and that would be able to provide temporal sequencing and duration annotations of events at the clause level. This is another motivation for the corpus project discussed here.

1.4. Previous Work

There are two dimensions to the annotation task discussed here. One is concerned with theoretical approaches to tense and event description that are useful in theoretical linguistic studies, and that facilitate the understanding and theorizing about intra- and cross-linguistic semantics of tense and event logic. The other dimension is related to annotation standards and proposals suggested for time and event description augmentation of text corpora that can facilitate the generation of corpora for data-driven machine learning for NLP. In the following, each of these dimensions is discussed separately.
For annotation purposes, there are various theoretical candidates that appear appropriate from a perspective that aims at the maximization of annotated data and minimization of annotation mistakes when focusing on events and time properties. Annotation of common tense properties and related features using grammatical concepts like present, past, and future is useful, but insufficient for the tasks at stake here, viz., event sequencing and interpretation of tense in complex sentences.
In the current approach and as part of the TIE-ML annotation schema, a variant of the Reichenbach Model (Reichenbach, 1947) has been adopted. The reason for selecting the Reichenbach model is explained in the following section. A discussion of the numerous other alternatives would be beyond the scope of this article.

1.4.1. The Reichenbach Model

Reichenbach (1947) introduces a theory of tense that provides the building blocks for developing a method to capture or describe time and event information in language. The theory presents three time variables: E for *event time*, R for *reference time*, and S for *speaker time*.
Event time refers to the time of the event in question. *Reference time* refers to a reference point or point of focus for that event, which can be expressed overtly via an adverbial such as *now* or *yesterday*, but can also be covert or implicit. *Speaker time* refers to the time of the utterance itself. These variables are ordered via two ordering relations with ',' denoting simultaneous, and '_' denoting separated sequencing.

(12) a. Simple Present (E,R,S where R = now)
I see Ross now.
b. Simple Past (E,R_S where R = yesterday)
I saw Ross yesterday.
c. Simple Future (S_E,R where R = tomorrow)
I will see Ross tomorrow.
d. Present Perfect (E_S,R where R = now)
I have seen Ross now.
e. Past Perfect (E_R_S where R = yesterday)
I had seen Ross yesterday.
f. Future Perfect (S_E_R where R = tomorrow)
I will have seen Ross tomorrow.

All three variables are explicit in each example above, regardless of whether perfect aspect is present. Note, however, that if we consider examples without perfect aspect—i.e., consider examples with simple past, simple present, or simple future tenses—only two of the three variables, speaker time S and event time E, are needed to distinguish the tenses. It is only when we introduce a second aspect category (here, the perfect) that we need a third time variable, reference time R, to distinguish between the resulting tense-aspect combinations. The TIE-ML annotation schema incorporates this variant of the Reichenbach model. See Comrie (1985) for a related variant of the Reichenbach model where reference time only appears in relative tenses.
The hypothesis in this project is that scope relations and other temporal properties such as tense, aspect, and temporal adverbials affect ordering relations between the variables expressed in the Reichenbach model. For example, where using a concrete temporal expression that anchors the predicate's reference time does not affect the relation between speaker time and event time, other properties such as scope relations, tense, and aspect from a dominating matrix clause may alter the Reichenbach variable ordering of subordinate clauses. To be able to study such effects cross-linguistically and over large corpora, as well as develop efficient models for the analysis or generation of tense and event-related expressions, the granularity in this project's approach to capturing temporal information has been extended from simple labels like *past*, *present*, and *future* to these Reichenbach variables.
The enterprise of offering a fine-grained annotation schema that entails scope relations and other temporal properties has been undertaken by Pustejovsky et al. (2003b) and their Time Markup Language project. In the next section, we introduce the project, discuss its characteristics and potential limitations, especially regarding the complexity of its annotation schema, and how TIE-ML offers a good compromise between details of annotations and limited effort.

1.5. TimeML and Annotation Standards

TimeML (Pustejovsky et al., 2003a; Pustejovsky et al., 2005) is an XML-based markup language and metadata standard developed for annotating events and temporal expressions in natural language or time information in general. It is the most detailed and theoretically grounded framework.

Historically, TimeML seems to have its roots in the workshop Time and Event Recognition for Question Answering Systems (TERQUAS) in 2002. At the TERQUAS workshop, recommendations for the enhancement of question answering systems were developed. Pustejovsky et al. (2003b) proposed TimeML for the annotation of events and temporal relations integrating for example the TIMEX2 tag (see for example Wilson et al. (2001)), a proposed inline XML tag with six attributes, and various other suggestions, along with other emerging schemata (Katz and Arosio, 2001). See Cavar et al. (2021) for a more detailed overview of the different standards and how they relate to TimeML.

TimeML is concerned with two major objectives. The first is to map predicates to events. The second is to establish a relative ordering between events.

In TimeML there are separate annotations for events and temporal expressions, and the specific anchoring or ordering dependencies are expressed in language. It provides four core annotation tags, i.e., EVENT, TIMEX3, SIGNAL, and LINK. EVENT encodes events that are punctual or that have a duration associated with them. The SIGNAL tag can be used to mark up function words with a temporal reference. Relationships between events are encoded via LINK tags.

Each of these tags provides sophisticated annotation properties that can cover complex events and temporal relations. The complexity of TimeML can be seen when considering for example the EVENT tag. It is broken down into types of events like *Reporting*, *Perception*, *Aspectual*, *I Action*, *I State*, *State*, and *Occurrence* events.

TimeML introduces not just new extensions to the TIMEX2 tag via new attributes. It also introduces temporal functions to allow intentionally specified expressions like *five months ago* or *in five days*. It allows for the annotations of SIGNALS that are relevant for the interpretation of temporal expressions, like temporal prepositions (e.g., *at*, *on*, *during*, *for*) or connectives (e.g., *while*, *after*, *before*). Event expressions that can be specified include a rich set of types like tensed verbs (e.g., *has left*, *was captured*, *will resign*), stative adjectives (e.g., *landed*, *sunken*, *stalled*), or event nominals (e.g., *destruction*, *merger*, *Military Operation*, *Gulf War*). It provides instruments to express dependencies between events and times, for example *anchoring*, *embeddings*, or *orderings*.

Although these sophisticated instruments facilitate the annotation of extremely detailed temporal information in language, their complexity requires extensive training of annotators to provide sufficient and useful data sets with acceptable annotation quality. We found the overall annotation process to demand significantly more effort than necessary when focusing on a subset of details related to event and time annotation required for our downstream machine learning applications.

The simplified TIE-ML schema presented in the next section aims to solve these issues while providing comprehensive annotations for time and event information that can easily be mapped and translated into the theoretically far superior annotation standard of TimeML

2. TIE-ML Standard and Approach

TIE-ML (Temporal Information Event Markup Language) (Cavar et al., 2021) is a simplified temporal annotation schema that focuses on event sequencing annotation and clause level temporal properties of main predicates. The goal of TIE-ML is to improve upon previous markup strategies' accuracy and productivity via simplification. Increasing the production of *good data* with the event and temporal properties annotated will facilitate the development of machine learning models for applications that can benefit from specific semantic analytics. This increase of productivity can also be achieved through simplifying the task for annotators.

Breaking tasks down to simple annotations of predicate tense, enumeration of events expressed by predicates, and labeling temporal expressions that encode duration or temporal anchoring simplifies the process, requires less training of annotators and reduces annotation errors.

TIE-ML was designed as an XML markup language that provides sentence and clause level annotations of text using the `S` and `C` tag respectively. While XML is a possible way to augment text with event and temporal information, the same approach can be achieved with a JSON-variant of TIE-ML. Alternative formats like CoNLL(-U)[1] (Buchholz and Marsi, 2006) can be generated as well.

TIE-ML focuses on the annotation of events expressed by individual predicates at the clause level. By enumerating each clause or independent predicate, an event is identified with an (`eventid`) as shown in the XML sample in Figure (1).

```
<tieml>
<s> <c eventid="1">
    Danny watched the movie
    </c>
    <c eventid="2">
    and ate popcorn
    </c>. </s>
<s> <c eventid="3">
    Josh brought the pizza
    </c>. </s>
</tieml>
```

Figure 1: TIE-ML example

While `eventid` reflects the presentation order of events in a text, temporal ordering is annotated by providing a `timeslot` identifier that reflects the relative

[1] See `https://universaldependencies.org/format.html` for a detailed explanation of the CoNLL-U format.

position of events on the time axis. For point-wise events, this is a concrete location of the event on the time axis. For events with an associated duration, this reflects the start point of the event on the time axis. These two properties are defined as attributes of the C (or S) tag in XML, as shown in Figure (2).

```
<s> <c eventid="1" timeslot="2">
    Before you fry the vegetables </c>
    <c eventid="2" timeslot="1">
    chop them into cubes
    </c>. </s>
```

Figure 2: TIE-ML timeslot example

In Figure ((2) the enumerated eventid does not correspond with the temporal order timeslot as reflected in their differing values amongst each respective clause.

The tense properties of the predicates are labeled using the Reichenbach variables *event time* E, *speaker time* S, and *reference time* R, which are introduced as XML attributes to the C tag at the clause level as shown in Figure (3). The values of these attributes are integers that reflect the relative order of coincidence or precedence, e.g., an S-value of 0 and an E-value of 0 represent present tense, an S-value of 0 and an E-value of -1 represent past tense, and an S-value of 0 and an E-value of 1 represent future tense.

```
<s> <c e="-1" s="0">
    Danny watched the movie.
</c> </s>
```

Figure 3: TIE-ML Reichenbach variables simple example

Note that because Figure ((3) is a simple past tense sentence, the *reference time* R does not appear. Values of -1 and 0 for *event time* and *reference time* respectively correspond to the Reichenbach notation of E_S. Figure (4) presents an annotation example of a future perfect sentence in which *reference time* R does appear.

```
<s> <c e="1" r ="2" s="0">
    Danny will have watched the movie.
</c> </s>
```

Figure 4: TIE-ML Reichenbach variables perfect example

Values of 1, 2, and 0 for *event time*, *reference time*, and *speaker time* respectively correspond to the Reichenbach notation of S_E_R.

Concrete expressions of reference time in the clause are encoded as attributes using the reference attribute in the C-tag. as shown in Figure (5).

```
<s> <c reference = "Monday">
    Jacob visited his mother on Monday.
</c> </s>
```

Figure 5: TIE-ML reference example

This XML annotation schema is kept intentionally simple to allow for efficient annotation of sequencing of events, as well as temporal features of predicates and temporal expressions in each respective clause. To be precise, our prediction is that TIE-ML is more efficient and leads to cleaner results much faster than alternative annotation approaches both when it comes to creating a resource that provides annotations of predicate properties and clausal relationships in particular geared toward the development of machine learning models, as well as when it comes to the quantitative and qualitative study of intra- and cross-linguistic temporal properties. To validate our prediction, we decided to use the Penn Treebank (Marcus et al., 1999) as a base-corpus and augment the syntactic and functional annotations with the proposed event and temporal properties.

XML as such, however, was not a convincing data format for annotators to work with or produce, even though powerful XML editors and tools can simplify the editing task tremendously. Instead of using XML as the annotation format, we decided to use TIE-ML XML as an exchange and conversion format, one that can be generated from formats provided by sophisticated annotation tools like INCEpTION, or one that can be converted into the extremely powerful TimeML annotation format.

In the following, we describe the adaptation and use of a specific configuration of INCEpTION for the TIE-ML style of annotation.

2.1. Annotation Implementation using INCEpTION

The INCEpTION platform (Klie et al., 2018) served for the annotation of the Penn Treebank corpus using the TIE-ML standard. The main reason for using it was the expectation that the annotation effort could be simplified and facilitated even more. In addition to providing excellent annotation instruments, INCEpTION also offers advanced management of annotators, corpora, and statistical tools for inter-annotator agreement analysis.

The specification of the three specific layers of annotations, namely: predicate, clausal, and temporal named entity annotations, is straightforward in INCEpTION. The predicate annotation layer is used to annotate predicates (whether finite or non-finite) and has an *Aspect* feature and a *Tense* feature.

The *Aspect* feature includes the following tags: *Simple*,

Progressive, *Perfect*, *Perfect Progressive*. In addition, annotation of *Voice* features is provided in form of an optional *Passive* tag.
The *Tense* feature includes the following tags: *Present*, *Past*, and *Future*. These tags are converted in the backend when transferring the annotations to the TIE-ML XML format to the corresponding Reichenbach variables.
The Temporal named entity annotation layer is used to mark temporal referents which anchor a given clause in a specific point in time. It includes the following tags: `TEMP` (for a temporal element in its pure form), `TEMPderiv` (for a temporal element as a derivational element), and `TEMPpart` (for a temporal element as part of a bigger token). These tags corresponds to the TIE-ML `reference` tag.
In periphrastic tense forms, the temporal cues are expressed as a sequence of verbal elements, i.e., auxiliaries and verbs. In copula constructions, the copula element and an additional adjectival or nominal head form the predicate of a clause. As an additional problem, Multi-word expressions in periphrastic tense forms can be realized discontinuously in a clause, as shown in example (13). To annotate the properties of the predicate in clauses with such discontinuities, each element of the predicate is labeled with the full predicate feature set independently, while adjacent sequences of lexical items are marked as one multi-word predicate unit.

(13) *John **was** mostly **reading** newspapers.*

Finally, for non-finite verb forms or for cases in which the copula or auxiliary is missing as shown in example (14), the verbal element that is overt is only labeled for Aspect (Simple, Perfect, Progressive, or Passive) and not for Tense since the tense marking is present on the auxiliary/copula.

(14) [*John is reading a book*] *and*
[____ __ *drinking tea.*]

The core annotation assumption is that each clause has only one core predicate. In some cases, this predicate can be opaque, for example, due to *ellipsis* or *gapping* applied to the clause or sentence (see for example Johnson (2008)). Opaque predicates are not yet annotated in this version of the corpus.
The individual lexical items are independently labeled with a part-of-speech tag in the Penn Treebank, which allows for automatic detection of inversion and deviation from canonical word order. This is relevant for the annotation of languages that allow for auxiliary verb inversions, for example, German (see also *VP topicalization* in Haider (1990)) or Croatian (Cavar and Wilder, 1994).
The clausal annotation layer is used to annotate clause boundaries. Each clause can be identified as main- or subordinate clause, including differentiation between complement, adjunct, or relative clauses. Clausal features that can be used in the annotation include the following:

- Clause ID (unique integer per clause within one sentence)
- Time Slot of an event (sequence of events using integer enumeration)
- Speaker Time
- Event Time (determines the tense of each clause predicate)
- Reference Time expressed by a tagged Temporal Named Entity in the clause (optional)
- Level of Embedding (integer indicating the depth of embedding of a clause in a sentence)
- Selected by ID (the ID of the clause containing the predicate that selects the clause, if the clause is a selected complement)

Each clause is given a clause ID corresponding to the TIE-ML `eventid` tag, and a Time Slot corresponding to the TIE-ML `timeslot` tag). The Speaker Time and Event Time correspond to the TIE-ML `s` and `e` tags and are determined for the main clause based on the characterization of the main clause's predicate and its *Tense* and *Aspect* tags.The Speaker Time and Event Time of a relative, complement or adjunct clause depend on those of the main clause. The Level of Embedding and Selected by ID features relate to the clausal hierarchy. The Level of Embedding of each main clause is 0 while that of each complement, adjunct, and relative clause is always 1 more than the clause they depend on. Relative, adjunct, and main clauses are by definition not selected elements, thus the *Selected by ID* label should always be 0, while in the case of a complement clause, the label should reflect the ID of the selecting clause.
Each clause contains a list of lexical items. Clauses can be rendered discontinuous within a complex sentence as in a (15). Segments of tokens with the same clause ID are assumed to be parts of the same clause.

(15) **Which car** did John say **that Mary will like** ?
clause_1 clause_2 clause_1

To exemplify the annotations we will use an export format for our data set. We utilize an interim format data exchange format from INCEpTION to Machine Learning algorithms to encode sentences and clauses similar to the CoNLL[2] Tab Separated Values (TSV) format. We separate sentences with an empty line and encode clauses by line, followed by tab-separated clause ID and time slot assignment. The enumeration of clauses starts with 1 for each sentence and it is expressed in the

[2]See `https://www.clips.uantwerpen.be/conll2006/`.

second column. The temporal order of the predicates (per clause just one) is encoded as the time-slot (TS) in the third column. The clauses are tokenized. The filenames correspond to the Penn Treebank filenames.

```
CLAUSE                        ID   TS
Which car                      1    2
did John say                   2    1
that Mary will like?           1    2

She will like the blue car.    1    1
```

This format is only one of many possible export formats that we generate from the INCEpTION output or storage format.
Using annotation IDs as in this case, it is possible to capture different types of very common discontinuities or dislocations in syntax and their relevance for semantic interpretation. This way it is also possible to cope with covert or incomplete predicates that are semantically implied in clauses that are subject to ellipsis or to similar phenomena.
Since the underlying sentence collection for the first level annotation for English is based on the Penn Treebank, all the annotations from the treebank (e.g., part of speech tags and syntactic structures) are available in addition to the clause level segmentation, temporal features, and sequencing provided in this project. This syntactic information provides hierarchical and scope information that can be utilized in various automatic conversions or machine learning tasks.
Additionally, we developed our own tense annotator for various Indo-European languages for clause level tense annotation to validate the user accuracy.
Note, however, that our focus is on the annotation of predicate sequencing along the time axis using a simple enumeration strategy, and additionally, on the tense information of each individual clause given its context, scope of dominance relation to other tenses and potential temporal expressions.

3. Data and Corpora

The corpora, samples, and scripts are made available at the public TIE-ML GitHub repository:

`https://github.com/dcavar/tieml`

More documentation and information about the project can be found at the website of the NLP-Lab:

`https://nlp-lab.org/timeevents/`

The annotations for English based in the first version on the Penn Treebank are made available in the GitHub repository. The dataset covers the freely available 10% of the Penn Treebank that are distributed in the Natural Language Toolkit (NLTK)[3] (Bird et al., 2009) data set. The full Penn Treebank annotation is available as a script that reads an existing treebank data set and generates the TIE-ML annotations.
A full linked data set for the treebank will be generated and made available in the TIE-ML GitHub repository. Samples from different languages and other tools are accessible there as well, including the INCEpTION to TIE-ML XML conversion script.
Contributions from volunteers and other teams or individuals are welcome. Please use GitHub pull requests as an instrument, and feel free to contact the NLP-Lab team.
Similar datasets are being developed for Arabic, Korean, Croatian, and other languages.

[3]See `https://www.nltk.org/` for more details.

3.1. Copyrights

The code produced by this project is shared in the public GitHub repository under the Apache License Version 2.0.
All texts and corpora in the public GitHub repository are licensed under the Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license, or – in case of third party data – under the specific license of the copyright holder, as noted in the README or LICENSE file in the corresponding sub-folder.

4. Results

For the evaluation of the claim that the TIE-ML annotation approach and utilization of INCEpTION the output and annotation quality per annotator can be maximized, we took the 10% portion of the Penn Treebank in the NLTK data set. This part consists of 199 files from the treebank, with a total of 3914 sentences and 93838 tokens. The number of clauses and overt main predicates will be updated here for the final paper when the validation of the annotations is complete and approved.

num. files	199
num. tokens	93838
num. sentences	3914

Table 1: Properties of the Pen Treebank Portion Annotated

For annotation, we used the adapted INCEpTION interface. The current number of annotators is 9. The scores in Table (2) reflect the current average while the 9 annotators have processed different sections of the corpus. The majority of the annotators are students in computational linguistics and linguistics at Indiana University - Bloomington, with varying experience and basic training in the syntax and semantics of events and temporal relations.
The time for annotation at each different level is given in table 2. This table reflects the average time scores after a first annotation round over a corpus sample.
The scores in Table (2) reflect the complexity of annotations of clause features such as selection relations, hierarchical depth, and time-slot assignment. Clause

Annotation type	Avg. time
Predicate labeling	13 sec./predicate
Clause boundaries	8.8 sec./clause
Clause features	50 sec./clause
Sentence	122 sec./clause

Table 2: Average Annotation Times by Annotation Type

boundary markup and predicate tense labeling are the fastest processes in the described setting. Note that clause boundary and predicate labeling also involves enumeration of events and sequencing, that is, assigning Reichenbach features to clause-level predicates and sequencing of events can be achieved with the described approach efficiently.

5. Conclusion

Overall, TIE-ML, a simple annotation schema focused on event sequencing annotation through the incorporation of Reichenbach variables, has been presented, and initial experiments with TIE-ML annotation using INCEpTION as a graphical front-end have proven to be highly informative.

Although, in the current stage, the main focus was on enriching existing English corpora, our ultimate goal is to report on intra- and cross-linguistic insights in hierarchical interpretation of tense and event sequencing in different language types (e.g., SVO, SOV, VSO), with variation in placement of embedded clauses and predicates (e.g., placement position and interpretation of adjunct vs. complement clauses).

In addition, generating large data sets using TIE-ML will provide the necessary data for training machine learning models in downstream event labeling applications that are able to guess the temporal sequencing and relation of events.

Finally, this project will apply the simple annotation scheme for temporal sequencing with some simple tweaks also to temporal duration annotation of events. By breaking the annotations down into simple tasks, we expect to improve the quantitative and qualitative properties of our resulting data sets. Temporal sequencing and duration are essential for commonsense reasoning models, which fall in the core focus of our research interest.

6. Acknowledgements

We are grateful to all the team members of the Indiana University NLP-Lab (`https://nlp-lab.org/`). Special thanks go to Steven Franks, Thomas Grano, Larry Moss, and Zoran Tiganj for helpful discussions, suggestions, and comments related to theoretical issues, approaches, and the project goals.

Some excellent suggestions and comments from reviewers will be worked into subsequent publications and material, and summarized on the project website. Due to time limitations it was not possible to take some of the suggestions into account and integrate them in this version of the article.

7. Bibliographical References

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.

Buchholz, S. and Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City, June. Association for Computational Linguistics.

Cavar, D. and Wilder, C. (1994). Clitic third in croatian. In H. van Riemsdijk et al., editors, *Clitics: Their Origin, Status and Position*, volume 6 of *Eurotype Woking Papers, Theme Group 8*. Mouton de Gruyter, Berlin.

Cavar, D., Dickson, B., Aljubailan, A., and Kim, S. (2021). Temporal information and event markup language: TIE-ML markup process and schema version 1.0.

Comrie, B. (1985). *Tense*, volume 17. Cambridge university press.

Haider, H. (1990). Topicalization and other puzzles of german syntax. In G. Grewendorf et al., editors, *Scrambling and Barriers*, pages 93–112. Benjamins, Amsterdam.

Johnson, K. (2008). *Topics in ellipsis*. Cambridge University Press, Cambridge; New York.

Katz, G. and Arosio, F. (2001). The annotation of temporal information in natural language sentences. In *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing*.

Kiparsky, P. (2002). Event structure and the perfect. *The Construction of Meaning*, pages 113–136.

Klie, J.-C., Bugert, M., Boullosa, B., Eckart de Castilho, R., and Gurevych, I. (2018). The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., ferro, L., and Lazo, M. (2003a). The timebank corpus. *Proceedings of Corpus Linguistics*, pages 28–34, 01.

Pustejovsky, J., no, J. C., Robert Ingria, R. S., Gaizauskas, R., Setzer, A., Katz, G., and Radev, D. (2003b). Timeml: Robust specification of event and temporal expressions in text. Technical Report SS-03-07, AAAI.

Pustejovsky, J., Ingria, R., Sauri, R., Castaño, J. M., Littman, J., Gaizauskas, R. J., Setzer, A., Katz, G., and Mani, I. (2005). The specification language TimeML.

Reichenbach, H. (1947). *Elements of Symbolic Logic*. The Free Press, New York.

Wilson, G., Mani, I., Sundheim, B., and Ferro, L. (2001). A multilingual approach to annotating and extracting temporal information. In *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing*.

8. Language Resource References

Marcus, M. P., Santorini, B., Marcinkiewicz, M. A., and Taylor, A. (1999). Treebank-3 ldc99t42. Web Download.