

TimeML-strict: clarifying temporal annotation

Leon Derczynski[♡], Hector Llorens^{♣◇}, Naushad UzZaman^{♠◇}

[♡]: leon@dcs.shef.ac.uk University of Sheffield, UK

[♣]: hlllorens@dlsi.ua.es University of Alicante, Spain

[♠]: naushad@cs.rochester.edu University of Rochester, USA

[◇]: www.nuance.com Nuance Communications

Abstract

TimeML is an XML-based schema for annotating temporal information over discourse. The standard has been used to annotate a variety of resources and is followed by a number of tools, the creation of which constitute hundreds of thousands of man-hours of research work. However, the current state of resources is such that many are not valid, or do not produce valid output, or contain ambiguous or custom additions and removals. Difficulties arising from these variances were highlighted in the TempEval-3 exercise, which included its own extra stipulations over conventional TimeML as a response.

To unify the state of current resources, and to make progress toward easy adoption of its current incarnation ISO-TimeML, this paper introduces TimeML-strict: a valid, unambiguous, and easy-to-process subset of TimeML. We also introduce three resources – a schema for TimeML-strict; a validator tool for TimeML-strict, so that one may ensure documents are in the correct form; and a repair tool that corrects common invalidating errors and adds disambiguating markup in order to convert documents from the laxer TimeML standard to TimeML-strict.

1 Introduction

TimeML (Pustejovsky et al., 2005) is an annotation scheme for the challenging task of annotation of temporal information over natural language text. Time as expressed in language is complex and often ambiguous, and determining how to annotate it for e.g. computational processing is accordingly difficult (Jaszczolt, 2009).

Almost a decade on from its release, the TimeML schema has been adopted by hundreds of projects worldwide, and has developed into an ISO standard (Pustejovsky et al., 2010). It is a comprehensive, expressive annotation markup language for temporal annotation, having been applied to significant amounts of resources and provided a framework for notably furthering research in temporal information extraction and understanding of temporal semantics. Indeed, machine-readable temporal annotation has found applications in a wide variety of domains, including: legal (Howald, 2011; Ramakrishna et al., 2011), linguistic theory (Derczynski and Gaizauskas, 2013), question answering (Saquete et al., 2009; UzZaman et al., 2012a), social media and data man-

agement (Derczynski et al., 2013), human interface design (UzZaman et al., 2011), sports coverage (Borg, 2007), transport accident analysis (Johansson et al., 2005), working with deaf children (Arfé et al., 2009), and especially the clinical (Jung et al., 2011; Sun et al., 2013).

Over time, use cases have been discovered where it is beneficial to make some voluntary constraints regarding how TimeML is used. Certain informal agreements were entered into by researchers who wished to exchange data reliably. Recently, these constraints have been formally applied in the 3rd international temporal evaluation exercise, TempEval-3 (UzZaman et al., 2012b), where the corpus¹ followed such agreements. This paper details and crystallises these constraints into a voluntary, formal standard describing a subset of TimeML, called TimeML-strict.

As well as serving as a canonical reference of the standard used in TempEval-3, this paper also introduces arguments for TimeML-strict, and a range of supporting tools. We describe a schema for TimeML-strict; a validation tool for checking whether a TimeML document is conformant; and a repair tool, for tightening up existing data such that it is compliant. Our hope is that compliant data becomes easier to automatically process, thus widening the potential user base of TimeML as well as easing interoperability (Lee and Romary, 2010). The resources presented should also aid automatic conversion of legacy temporal annotation to ISO-TimeML, increasing the useful lifetime of community linguistic resources that represent large amounts of effort and investment in semantic annotation.

The paper is structured as follows. Section 2 discusses common difficulties encountered by end-users of TimeML. Section 3 describes the changes in TimeML-strict. Section 4 describes supplementary resources and Section 5 makes explicit some things that TimeML-strict does not address. We conclude in Section 6.

2 Problems with the current state of resources

What follows is a selection of common technical issues encountered when processing TimeML, followed by a requirements specification for a standard clarification.

2.1 Validity

As an XML format, TimeML documents can be validated by means of an XML “document type description”, which is part of the TimeML standard. However, not all resources currently distributed as TimeML are valid according to their XML document type description (DTD). This gives many problems when loading data as XML, via e.g. the document object model (DOM). The inability to load TimeML documents as XML directly removes many of the advantages of choosing XML for the standard – such as ease of use with text processing frameworks (Ogren et al., 2008; Cunningham et al., 2013) – and costs many man-days a year per researcher working with invalid data. Errors range from mis-typed element named to wrongly-encoded characters (from e.g. other alphabets) to SGML-valid but structurally inconsistent documents (e.g. where

¹Available in the ACL Data and Code Repository, reference [ADCR2013T001](#)

two TIMEX3 annotations have the same “unique” ID). Many different tools have made parallel efforts to overcome these difficulties. However, all these efforts would not be required if content creators published valid data in the first place.

2.2 Timestamps

It is sometimes unclear what the document creation time is. TimeML’s TIMEX3 functionInDocument attribute is there to label whether a timex is publication date, creation date and so on – an expressive and useful part of the schema, permitting capture of multiple important dates which often act in two roles, both as timexes in discourse and as document meta-information. Often, only one of these special-function dates is specified (creation time or publication date), though sometimes none is, and sometimes more than one is. However, there is no way of defining which of these dates should be used as the default reference time. Having such a definition is critical to timex normalisation (Llorens et al., 2012) as well as to accurate replication of results.

2.3 What to annotate

TimeML is often used to annotate newswire documents (e.g., in the biggest TimeML corpus, TimeBank (Pustejovsky et al., 2003); in the TempEval-3 corpus; and also in the AQUAINT TimeML corpus). A lot of these are taken verbatim from the source, and include a preamble of metadata that is essentially gibberish – certainly not natural language (Section 3.2 below contains an example). This non-linguistic content often gets in the way of working with various NLP tools: it is often unclear how this preamble should be treated. Does one count it as a single sentence? Should headlines and editorial comments within it be annotated? How about numerically encoded dates that occur as fragments?

2.4 Inconsistencies

Many documents are produced that appear consistent but are difficult to process. These may include, for example, non-standard id labels; EVENT elements should always have an eid attribute that takes the form of eXX where XX is a positive integer, though some tools treat the field as freeform text, or omit the e. In other cases, edits to a document may create partial information (through both deletion and insertion of annotations). For example, after deleting an event instance, an ALINK may still use that event instance ID as one of its arguments. All these phenomena create speed bumps working with TimeML, but can be readily checked for.

2.5 Requirements

The majority of problems above stem from the organic way in which tools and resources have sprung up over the past years, all using the framework of TimeML. We hope to provide a common ground – and means of reaching it – that is both TimeML-compatible and also very easy to work with. One goal of this standard refinement is

to ease the process of programming with TimeML. Therefore, it should be easy to implement systems that accept TimeML-strict. For acceptance of TimeML-strict to be a sufficient programming requirement for working with TimeML, it is important that legacy data and annotations produced by legacy tools are compatible with systems that expect TimeML-strict. Finally, one must not constrain the expressiveness of TimeML: rather, the goal is to carefully preserve this expressiveness, and maximise access to TimeML-annotated resources.

3 Changes new in TimeML-strict

This section details our proposed annotations, as an addendum to the TimeML standard v1.2. Parts of ISO-TimeML are accepted either, though the standard is not as well-used as TimeML, and partially in a state of flux. TimeML-strict does include some features of ISO-TimeML: most notably,

- instantiating events by including `eiid` and other event instance attributes within an `EVENT` label, thus eliminating the need for `MAKEINSTANCES` for the majority case where events are instantiated once;
- support event verb form and predicate attributes (`vForm` and `pred`).

3.1 The `DCT` element

This is introduced to resolve potential ambiguity regarding the document's creation time, that should be used as a default anchor for `timexes` within the document. There must be exactly one `DCT` per document. This element should enclose a single `TIMEX3` element, with no other intervening nodes (including text nodes) – see Example 1.

```
(1) <DCT><TIMEX3 functionInDocument="CREATION-TIME"
    temporalFunction="false" tid="t0" type="DATE"
    value="2013-03-22">March 22, 2013</TIMEX3></DCT>
```

In the case of documents where `DCT` is not known, not given, or otherwise unclear, give an underspecified self-closing day-level `timex` annotation (Example 2).

```
(2) <DCT><TIMEX3 tid="t0" value="XXXX-XX-XX" /></DCT>
```

3.2 The `TEXT` element

This is used to specify exactly the bounds of linguistic content in the document. The goal is to be clear which content should be considered for annotation, and allow exclusion of non-linguistic document content. Example 3 shows how this element can be used to exclude newswire preamble.

```
(3) <TimeML>
    AP900815-0044
```

```

AP-NR-08-15-90 1337EDT
u i PM-GulfRdp 8thLd-Writethru 08-15 1334
PM-Gulf Rdp, 8th Ld-Writethru,a0605,1368
Saddam Seeks End To War With Iran; Bush To Urge Jordan To Close
Port
Eds: SUBS 28th graf pvs, Crown Prince... to CORRECT spelling of
Hassan; pick up 29th graf pvs, 'A CBS...'
LaserPhotos WX6,7,XSAV1,NY5,10,TOK1,XAAFB1,AMM1, LaserColor XAAFB1
By CHRISTOPHER BURNS
Associated Press Writer

<TEXT>
    Iraq's Saddam Hussein, <EVENT eid="e5"
class="STATE">facing</EVENT> U.S. and Arab troops at the Saudi
...
</TEXT>

</TimeML>

```

Each document must have exactly one TEXT element. No particular XML hierarchical relation between the TEXT and DCT elements is required; DCT may be before, after, within, or even contain the TEXT element (i.e. parent, child and sibling are all fine).

3.3 Schema validation requirement

All TimeML documents should include a reference to the TimeML DTD, in order to assist with their validation. This requirement has not been sufficient to ensure clean, legible XML. TimeML-strict documents *must* be valid.

As well as TimeML DTD validation, TimeML-strict also requires that documents validate according to a more rigorous XML schema: the TimeML-strict XSD. This is the core strict requirement, intended to aid processing of TimeML by other tools. Schema compliance make transforming TimeML-strict to other formats (such as those following the ISO Language Annotation Framework, including ISO-TimeML) easy, with one-off transformation descriptions via e.g. XSLT. Following the standard enables easy conversion to ISO-TimeML whenever updates are released to the community (through formal XML translations, instead of text processing or intricate SAX event handlers and so on).

A TimeML validation tool is made available in order to assist meeting this requirement, and the corresponding XML schema file (see [Section 4.2](#)).

3.4 No phantom element IDs

XML schemas can enforce checks for missing references and elements. For example, a TLINK may reference a nonexistent event instance; elements may have malformed identifiers. TimeML-strict requires that every reference to an element be an identifier giving reference to a findable element.

3.5 Semantics of DURING

In his 1983 account, Allen determined a small set of distinct possible relations between two temporal orderings (Allen, 1983). TimeML uses this full-interval temporal relation set for TLINKs. For the most part, the relations in Allen’s paper and in TimeML are simple to match. TimeML introduces the IDENTITY relation, to distinguish between events (or times) that happen at the same time, and those that are also the same thing. However, Allen’s `overlap` and `overlap-inverse` relations do not seem to be accounted for in TimeML. Similarly, the TimeML DURING / DURING_INV relations are not defined strongly in the annotation guidelines or v1.2 specification.

The apparent absence of Allen’s `overlap` relations leaves a hole in TimeML’s expressiveness. See Example 4 – there is no other temporal relation that would be appropriate between `e1` and `t1`. The winter starts within 2012, but continues beyond its termination. The relation is not inclusion or simultaneity, and both BEGINS and ENDS (and their inverses) require a shared interval endpoint, which is also not the case here.

- (4) The `<EVENT eid="e1" eiid="e11" class="OCCURRENCE">winter</EVENT>` that started in `<TIMEX3 tid="t1" type="DATE" value="2012">2012</TIMEX3>` was one of the coldest on record.

Further, the DURING relation seems to duplicate functionality or either INCLUDES or SIMULTANEOUS, depending on subjective interpretation. One popular definition is that it should be used when an event occurs during a timex. However, this functionality is already explicitly provided by SIMULTANEOUS / IS_INCLUDED. Choosing one of those gives a more precise relation. It is unclear why one would want to omit Allen’s `overlap` functionality but include an underspecified superlabel for just one specific circumstance. Also, the idea that some relation labels can only apply to particular types of intervals seems unintuitive, and is a departure from the general theme in TimeML of abstracting events and timexes to intervals.

Finally, it is difficult to introduce a new `relType` value to TimeML for these relations; doing so breaks TimeML compatibility, and signifies a departure from TimeML, which is the opposite of our goal.

In TimeML-strict, this apparent mismatch is interpreted as an oversight, and the two explicitly mapped as follows.

1. TimeML DURING is equivalent to Allen’s “`overlap inverse (oi)`”: A DURING B is read as “*A starts during B and persists beyond the end of B*” (e.g. `overlap` where A starts within B);
2. TimeML DURING_INV is equivalent to Allen’s “`overlap (o)`”: A DURING_INV B is read as “*During A, B starts and persists beyond the end of A*” (e.g. `overlap` where B starts within A).

Under TimeML-strict, for the intervals in Example 4, one might annotate:

```
<TLINK lid="l1" eventInstanceID="e11" relType="DURING"
  relatedToTime="t1" />
```

This change is perhaps in conflict with some prior interpretations of TimeML, but it is the only way in which TimeML's relation types can be made to cover the full set of interval relation configurations and removes two arguably redundant links, while staying valid.

As this change risks taking a departure from some conventions, one should pay regard to impact on existing resources. Brief examination of TimeBank 1.2 suggested annotator ambiguity between all of the INCLUDES, SIMULTANEOUS and DURING link relation types. We do not specify how to deal with such annotation, and recommend that annotations (especially in resources) be treated as slightly fuzzy, as per the TimeML recommendation.

Existing closure tools should be readily modifiable to meet this specification change and indeed some have already been using this interpretation for a few years.

4 Resources provided

With TimeML-strict, three resources are made available for working with documents and the specification.

4.1 XML Schema Definition

Core to TimeML-strict is a formal validation schema. This is substantially different from TimeML 1.2's XML schema definition (XSD), building in stricter checks. There are two key scenarios for this schema's use: to help TimeML producers be sure that they have generated shareable, legible TimeML; and to enable those trying to read TimeML to be aware of the expected breadth of expression and level of data consistency.

This schema includes TIMEX3 support. There is no single reference point for the TIMEX3 standard, but rather TimeML builds on earlier TIMEX standards. The TimeML-strict schema incorporates the TIMEX2 spec from [Ferro et al. \(2005\)](#) and the appropriate adaptations in TimeML 1.2. The schema is included with the TimeML validator.

4.2 TimeML validator

A Java validation tool that verifies whether or not documents are acceptable as TimeML-strict. For documents that are not valid, information is given to allow content creators to find the source of the problems. This tool is available via github.²

4.3 Automatic migration and repair of TimeML

To ease adoption of TimeML-strict, and to convert potentially invalid data into a consistent, valid format, a TimeML repair tool is made available. This converts and fixes common mistakes in older TimeML, and enables older resources to be processed by newer tools.

²See <https://github.com/hllorens/TimeML-validator>

The DCT and TEXT elements are automatically added by the tool, if possible, which also attempts to rectify several common mistakes such as invalidly-structured ID strings or references to missing entities. As TimeBank v1.2 isn't valid under TimeML-strict, this tool is absolutely critical. The tool has also been tested with a few common TimeML-generating systems and can rectify their output, repairing both invalid TimeML and also adding requisite information to make the output TimeML-strict compliant. It is accessible via open-source repository.³

5 Beyond the scope of TimeML-strict

This paper has so far considered problems encountered “in the wild” and proposed fixes for them. There are also those problems that are deliberately *not* addressed by TimeML-strict. This section introduces a few of these phenomena with explanations of why TimeML-strict does not constraint against them.

The extent of timex and event elements should be just a single word, according to the TimeML English annotation guidelines. However, there are many cases where this is not possible, especially with timexes, where qualifying words are critical (e.g. *last* in “*last year*”). Also, some resources migrated from older standards include slightly longer words (Derczynski et al., 2012). Making this invalid in TimeML-strict would be an unreasonable constraint and would reduce the expressiveness of TimeML, while offering little benefit to the technical process (the annotated text is e.g. a single DOM node regardless of word count).

TimeML-strict also does not tackle temporal consistency. The transitive nature of many interval relation types means that it is possible to create an annotation that is inconsistent (Example 5).

(5)

```
<TLINK lid="l1" eventInstanceID="ei1" relType="BEFORE"
      relatedToEventInstance="ei2" />
<TLINK lid="l2" eventInstanceID="ei1" relType="INCLUDES"
      relatedToEventInstance="ei2" />
```

Although this presents difficulties when performing temporal closure or attempting to use inference-based learning for relation labelling, such annotations remain valid. TimeML should be capable of annotating any document. In the case of newswire, one should not make the assumption that the linguistic utterances of journalists are temporally consistent in the first place.

It is also possible to create somewhat “orphaned” annotations, e.g. uninstantiated events. There is no requirement to instantiate events, and inclusion of any event attribute other than its class is optional. Indeed, TimeML’s possible values for part of speech, tense, aspect and so on may be viewed as recommendations rather than declarations of the best way to annotate this values for temporal information processing, reducing how critical this information is.

Some of these kinds of meta-consistency is explicitly checked for by the CAVaT tool Derczynski and Gaizauskas (2010), which includes a selection of modules for validating TimeML. However, they are all permissible under TimeML-strict.

³See <https://bitbucket.org/leondz/timeml-repair>

6 Conclusion

This paper has detailed a refinement of the TimeML standard, hoping to bring together the many diverse outputs of the successful ongoing TimeML project and make life easier for those working with TimeML data. Along with the abstract parts of the refinement, three concrete tools are presented: a schema; a validation tool; and a repair tool. This is an iterative, voluntary step, which has the added benefit of preparing for rapid and seamless transition between TimeML and ISO-TimeML. TimeML-strict offers a common base for simpler computing with temporal annotations, allowing interested researchers to get on with experimentation and discovery.

Acknowledgements

The authors would like to thank James Pustejovsky, Marc Verhagen and Rob Gaizauskas for helpful discussions over the past years. We also thank the TempEval-3 participants (UzZaman et al., 2013) for their detailed feedback and for being superlative test subjects. Part of this work was supported by CHIST-ERA EPSRC grant No. EP/K017896/1 uComp,⁴ and also by Aarhus University, Denmark who kindly provided facilities.

References

- Allen, J. F. (1983), “Maintaining knowledge about temporal intervals.” *Communications of the ACM*, 26, 832–843.
- Arfé, B., R. Gennari, and O. Mich (2009), “Evaluations of the lode temporal reasoning tool with hearing and deaf children.” In *Cognitive and Metacognitive Educational Systems (MCES2009)*, AAAI Fall Symposium.
- Borg, M. (2007), “Time extraction from real-time generated football reports.” *Projektarbeten 2006*, 15.
- Cunningham, H., V. Tablan, A. Roberts, and K. Bontcheva (2013), “Getting More Out of Biomedical Documents with GATE’s Full Lifecycle Open Source Text Analytics.” *PLoS computational biology*, 9, e1002854.
- Derczynski, L. and R. Gaizauskas (2010), “Analysing Temporally Annotated Corpora with CAVaT.” In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, 398–404.
- Derczynski, L. and R. Gaizauskas (2013), “Empirical Validation of Reichenbach’s Tense Framework.” In *Proceedings of the 10th International Conference on Computational Semantics*, 71–82, Association for Computational Linguistics.

⁴See <http://www.ucomp.eu/>

- Derczynski, L., H. Llorens, and E. Saquete (2012), “Massively Increasing TIMEX3 Resources: A Transduction Approach.” In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, 3754–3761, European Language Resources Association (ELRA).
- Derczynski, L., B. Yang, and C. S. Jensen (2013), “Towards context-aware search and analysis on social media data.” In *Proceedings of the International Conference on Extending Database Technology*, 137–142, ACM.
- Ferro, L., L. Gerber, I. Mani, B. Sundheim, and G. Wilson (2005), “TIDES 2005 standard for the annotation of temporal expressions.” Technical report, MITRE.
- Howald, B. S. (2011), *The transformation of spatial experience in narrative discourse*. Ph.D. thesis, Georgetown University.
- Jaszczolt, K. (2009), *Representing time*. Oxford University Press.
- Johansson, R., A. Berglund, M. Danielsson, and P. Nugues (2005), “Automatic text-to-scene conversion in the traffic accident domain.” In *International Joint Conference on Artificial Intelligence*, volume 19, 1073, LAWRENCE ERLBAUM ASSOCIATES LTD.
- Jung, H., J. Allen, N. Blaylock, W. de Beaumont, L. Galescu, and M. Swift (2011), “Building timelines from narrative clinical records: initial results based-on deep natural language understanding.” In *Proceedings of BioNLP 2011 Workshop*, 146–154, Association for Computational Linguistics.
- Lee, K. and L. Romary (2010), “Towards interoperability of iso standards for language resource management.” In *Proceedings of the International Conference on Global interoperability for Language resources*.
- Llorens, H., L. Derczynski, R. Gaizauskas, and E. Saquete (2012), “TIMEN: An Open Temporal Expression Normalisation Resource.” In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, European Language Resources Association (ELRA).
- Ogren, P. V., P. G. Wetzler, and S. J. Bethard (2008), “ClearTK: A UIMA toolkit for statistical natural language processing.” In *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP*, 32–38.
- Pustejovsky, J., P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, et al. (2003), “The TimeBank corpus.” In *Proceedings of the Corpus Linguistics Conference*.
- Pustejovsky, J., B. Ingria, R. Saurí, J. Castano, J. Littman, R. Gaizauskas, A. Setzer, G. Katz, and I. Mani (2005), “The specification language TimeML.” *The Language of Time: A reader*, 545–557.

- Pustejovsky, J., K. Lee, H. Bunt, and L. Romary (2010), “ISO-TimeML: An International Standard for Semantic Annotation.” In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, European Language Resources Association (ELRA).
- Ramakrishna, K., V. Guda, B. P. Rani, and V. Chakati (2011), “A Novel Model for Timed Event Extraction and Temporal Reasoning in Legal Text Documents.” *International Journal of Computer Science and Engineering Survey*, 2, 39–48.
- Saquete, E., R. Munoz, and H. Llorens (2009), “Enhancing QA systems with complex temporal question processing capabilities.” *Journal of Artificial Intelligence Research*, 35, 775–811.
- Sun, W., A. Rumshisky, and O. Uzuner (2013), “Evaluating temporal relations in clinical text: 2012 i2b2 challenge.” *Journal of the American Medical Informatics Association*.
- UzZaman, N., J. P. Bigham, and J. F. Allen (2011), “Multimodal summarization of complex sentences.” In *Proceedings of the 16th international conference on Intelligent user interfaces*, 43–52, ACM.
- UzZaman, N., H. Llorens, and J. Allen (2012a), “Evaluating temporal information understanding with temporal question answering.” In *Proceedings of IEEE International Conference on Semantic Computing*.
- UzZaman, N., H. Llorens, J. F. Allen, L. Derczynski, M. Verhagen, and J. Pustejovsky (2012b), “Tempeval-3: Evaluating events, time expressions, and temporal relations.” *CoRR*, abs/1206.5333.
- UzZaman, N., H. Llorens, L. Derczynski, M. Verhagen, J. F. Allen, and J. Pustejovsky (2013), “SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations.” In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*.