The TARSQI Toolkit

Marc Verhagen and James Pustejovsky

Computer Science Department, Brandeis University, Waltham, USA {marc,jamesp}@cs.brandeis.edu

Abstract

We present and demonstrate the updated version of the TARSQI Toolkit, a suite of temporal processing modules that extract temporal information from natural language texts. It parses the document and identifies temporal expressions, recognizes events, anchor events to temporal expressions and orders events relative to each other. The toolkit was previously demonstrated at COLING 2008, but has since seen substantial changes including: (1) incorporation of a new time expression tagger, (2) embracement of stand-off annotation, (3) application to the medical domain and (4) introduction of narrative containers.

Keywords: TimeML annotation; tools; temporal processing.

1. Introduction

The TARSQI Toolkit (TTK) is a suite of temporal processing modules. It identifies temporal expressions and events in natural language texts, and parses the document to both order events and anchor them to temporal expressions. TTK was developed primarily from 2004-2008 in the context of the IARPA AQUAINT project, which aimed at building question answering systems an din which temporal information was considered a needed basic capability that would increase performance of question answering systems. Recently, development has picked up again and four major changes have been made or are being made:

- 1. replacement of an old and hard-to-maintain time expression tagger
- 2. adoption of Linguistic Annotation Framework standards
- 3. application to the medical domain
- 4. introduction of narrative containers

In this demo paper, we first give a short overview of TimeML, the representation language embodied in the toolkit. We then describe the current state of the toolkit, focusing especially on recent and current changes.

Further details on the toolkit and individual components are available in (Verhagen et al., 2005; Verhagen and Pustejovsky, 2008; Saurí et al., 2005; Saurí et al., 2006b; Verhagen, 2005; Saurí et al., 2006a; Mani et al., 2006; Mani et al., 2007) and on the TimeML website.¹

2. TimeML

TimeML (Pustejovsky et al., 2003a; Pustejovsky et al., 2005; Pustejovsky et al., 2010) is a specification language for events and temporal expressions. Recently, TimeML has been consolidated as an international cross-language ISO standard (ISO WD 24617-1:2007) within the semantic annotation framework.

Events in articles are naturally anchored in time within the narrative of a text. For this reason, temporally grounded events are the very foundation from which we reason about how the world changes. Without a robust ability to identify and extract events and their temporal anchoring from a text, the real aboutness of the article can be missed. Moreover, since entities and their properties change over time, a database of assertions about entities will be incomplete or incorrect if it does not capture how these properties are updated throughout the timeline.

There are four major data structures that are specified in TimeML: EVENT, TIMEX3, SIGNAL, and LINK. TimeML considers *event* a cover term for situations that happen or occur. Events can be punctual or last for a period of time. TimeML also considers as events those predicates describing states or circumstances in which something obtains or holds true. Some examples of events are given in example (1) below. Note that the situations that happen or occur may do so in a hypothetical future.

- (1) a. A fresh flow of lava, gas and debris **erupted** there Saturday.
 - b. Israel will ask the United States to delay a military **strike** against Iraq until the Jewish state is fully prepared for a possible Iraqi **attack**.
 - c. A Philippine volcano, **dormant** for six centuries, began exploding with searing gases, thick ash and deadly debris.

Temporal expressions in TimeML are marked up with the TIMEX3 tag and can, amongst others, refer to dates, times and durations. There are three major types of TIMEX3 expressions: (i) fully specified temporal expressions, which provide all the information necessary in order to identify the point or period of time they are referring to; e.g., *June 11, 1989*; (ii) underspecified temporal expressions, which require the use of some contextual information in order to interpret the point in time they are referring to; e.g., *early in the morning* and *Monday*, and (iii) durations, such as *three months*.

The SIGNAL tag is used to annotate sections of text, typically function words, that indicate how temporal objects (events and temporal expressions) are to be related to each other.

(2) They will investigate the role of the US **before**, **during** and **after** the genocide.

¹http://timeml.org/.

Finally, TimeML introduced the LINK tags. There are three types of LINK tags in TimeML, which encode the various relations that exist between the temporal elements of a document, namely, events and temporal expressions:

- 1. TLINK: a Temporal Link representing the temporal relationship holding between two events, or between an event and a time. The actual relation is encoded with a relType attribute whose values include BEFORE, AFTER, INCLUDES, IS_INCLUDED, and SIMULTANE-OUS.
- 2. SLINK: a Subordination Link used for contexts introducing relations between two events. One example is the relation between the reporting verb and the embedded event in *John said he drank some wine*.
- 3. ALINK: an Aspectual Link representing the relationship between an aspectual event and its argument event. An example is the relation between *finished* and *assembling* in *John finished assembling the table*.

The original version of the TARSQI Toolkit generates all tags mentioned above except for the SIGNAL tag. It also added a MAKEINSTANCE tag for events in order to deal with cases like *He taught on Monday and Wednesday*, where there are two instances of the teach event. This distinction is now made at the level of the EVENT tag itself and MAKEINSTANCE was eliminated from both TimeML and TTK.

3. Overview of the Toolkit

A diagram of the architecture of TTK is given in Figure 1 on the next page. Input text is first processed by the *DocumentModel*, which takes care of document-level properties like encoding and meta tags. The DocumentModel hands clean text to the other components which are allowed to be more generic.

The *PreProcessor* uses standard approaches to tokenization, part-of-speech tagging and chunking. The tokenizer and chunker are homegrown, but for tagging TTK provides an interface to the TreeTagger (Schmid, 1994).

BTime is a temporal expression tagger that recognizes the extents of time expressions and generates their normalized values, it will be discussed further in section 3.2.. *Evita* is a domain-independent event recognition tool that performs two main tasks: robust event identification and analysis of grammatical features such as tense and aspect.

Slinket introduces SLINKs and classifies them into the categories factive, counterfactive, evidential, negative evidential, and modal, based on the modal force of the subordinating event (Saurí et al., 2006b). Slinket uses a pattern library for a well-delimited group of verbal and nominal predicates such as *regret*, *say*, *promise* and *attempt* where the patterns identify the subordination context.

The temporal processing stage includes three modules that generate TLINKS.

1. *Blinker* is a rule-based component that applies to certain configurations of events and timexes, using predominantly syntactic cues.

- 2. *S2T* takes the output of Slinket and uses about a dozen syntactic rules to map SLINKs onto TLINKs. For example, one S2T rule encodes that in SLINKs with reporting verbs where both events are in past tense, the reporting event occurred after the event reported on.
- 3. The *TLink Classifier* is a MaxEnt classifier that identifies temporal relations between previously recognized events and times in a text. It is trained on the Time-Bank corpus (Pustejovsky et al., 2003b).

TLINKS generated by the three modules above are not guaranteed to be consistent with each other. The Link Merging stage uses a greedy algorithm to merge TLINKS into a consistent whole. First, all links are ordered on their confidence score and put into a queue. Currently these scores are either global or local. Global confidence scores are derived from the observed precision of the component that generated the links. For example, links generated by S2T are considered high precision and are always deemed more reliable than links generated by the classifier. Links generated by the classifier come with a confidence score assigned by the classifier and these scores are used to order all classifier links.

Merging proceeds by first creating a graph that contains all events and time expressions as nodes, but there are no constraints expressed on the edges. Those constraints are added by the temporal links that are popped off the queue and are added one by one to the graph. Each time a link is added a constraint propagation component named *Sputlink*, based on Allen's interval algebra (Allen, 1983), is applied. If a link cannot be added because it is inconsistent with the constraint already on the edge, then the link is skipped. The result is a consistent annotation where high precision links are preferred over lower precision links.

To present temporal relations visually, TTK uses a visualization scheme named TBox (Verhagen, 2007), which was added to the Tango TimeML annotation tool (Verhagen et al., 2006). It uses left-to-right arrows, box inclusion and stacking to encode temporal precedence, inclusion, and simultaneity respectively (see Figure 2).



Figure 2: The TBox Representation

In this representation, temporal relations are deterministically and unambiguously mapped to specific ways of drawing them. And vice versa, a particular way of positioning two events always indicates the same temporal relation. Note that vertical positioning does not imply any temporal relation.



Figure 1: TARSQI Toolkit Architecture

3.1. Adopting Stand-off Annotation

Originally, the TARSQI Toolkit added annotations as inline XML tags to the source document. Recently however it has become increasingly clear that there is value in separating tags from the primary data that they annotate. The Linguistic Annotation Format (LAF), a standard for linguistic annotation developed by the International Standard Organization (Ide and Romary, 2006; Ide and Suderman, 2007), provides guidance on the basic principles for representing linguistic annotations are separated from the data they annotate (that is, LAF requires stand-off annotation), (ii) annotation structure and content are separated, and (iii) mappings between annotation occur via a pivot format.



Figure 3: LAF Annotation Layers

The LAF data model for annotations comprises a directed graph referencing regions of primary data as well as other annotations, in which nodes are labeled with feature structures providing the annotation content. The graph is initiated by creating virtual nodes between all characters in the primary data. Then leaf nodes can be created by referring to spans defined by virtual nodes. For example, for the string *"The clock struck ten."*, we can create leaf nodes as follows (this example is taken from (Ide and Romary, 2006)):

<edge id="e1" from="0" to="3"/>
<edge id="e2" from="4" to="9"/>
<edge id="e3" from="10" to="16"/>
<edge id="e4" from="17" to="20"/>
<edge id="e5" from="20" to="21"/>

Typically, this first layer of annotation is referred to as the base segmentation. Other annotations can be added on top of this annotation. Below is a node from a layer defining lemmatization and parts of speech:

```
<edge id="t2" ref="e2">
  <fs type="token">
      <f name="lemma" sVal="clock"/>
      <f name="pos" sVal="NN"/>
      </fs>
</edge>
```

The toolkit has been adapted to use stand-off annotation throughout. Inline XML is still an output option, but by default TTK creates layered files that point to other tag identifiers in other layers or to character offsets in the primary data.

3.2. BTime – Time Expression Tagger

The 2008 version of TTK used a component that was basically a wrapper around the Tempex Perl module (Mani and Wilson, 2000), which used complex regular expressions to recognize the extent of time expressions as well as a set of rules to determine the normalized value of the TIMEX3. While performing at a reasonable level (which was then state-of-the-art), it was hard to make updates to the script.

BTime consists of two components. One components implements the Early algorithm (Earley, 1970) and uses a grammar of 82 context-free rules that can be defined declaratively, as in the example below which gives a fragment of one of the rules (many of the other disjuncts are left out).

```
date ->
  day month year {
    CalendarDate(_[2], _[1], _[0]) }
  | month day year {
    CalendarDate(_[2], _[0], _[1]) }
  | day month "," year {
    CalendarDate(_[3], _[1], _[0]) }
```

The normalized value of the time expressions is built during rule application by calling pre-defined constructors in the action part of the rule. In many cases, values thus constructed remain underspecified. The other BTime component uses temporal functions that take the underspecified TIMEX3 and create the normalized value given an anchor time, which is determined using a set of simple heuristics.

3.3. Application to the Medical Domain

Until recently, development on TimeML and the TARSQI Toolkit was heavily skewed towards the newswire domain. Under an exploratory NIH grant, several experiments have been undertaken to start adapting TTK to the medical domain. One experiment focused on a set of discharge summaries and the task of selecting those documents where the patient was taking statins at the time of the patient's admission to the hospital (Stubbs and Harshfield, 2010). In order to successfully extend TTK, two major changes needed to be implemented:

- 1. The TimeML notion of event was expanded to include medications, especially the implied period when a medication was taken.
- 2. Temporal information encoded in several headers in the discharge summary was taken into account.

External Perl scripts were created that (1) labeled statins as events using a dictionary of aliases, and (2) located section headers and associated temporal information with them. The TTK Blinker module was then modified to make sure that it uses temporal information in headers when determining anchoring of events in the section. With these additions, TTK performed with 84% accuracy on the document selection task.

While very preliminary, this experiment unveiled the major obstacles and presented ways to adapt the toolkit for another domain. More specifically, the system needed to allow for domain-dependent notions of event and the Evita module needs to be sensitive to this. In addition, the document structure needed to provide more guidance to the relation discovery of TTK.

3.4. Narrative Containers

Related to the issue of document structure above is the rhetorical structure of the document and how it informs the anchoring of events. Recent work on TimeML and Time-Bank has explored how the notions of Narrative Time and Narrative Container can be used to increase the informational content of an annotation (Pustejovsky and Stubbs, 2011).

A Narrative Time is the current temporal focus of the document, it is updated while the reader progresses through the document. The Narrative Container is an interval that extends into the past from the creation time of a document and that can be used to anchor recent events to. As an example, take the fragment below.

April 25, 2010 7:04 p.m. EDT-t0

S1: President Obama paid-e1 tribute Sunday-t1 to 29 workers killed-e2 in an explosion-e3 at a West Virginia coal mine earlier this month-t2, saying-e4 they died-e5 "in pursuit of the American dream."

S2: The blast-e6 at the Upper Big Branch Mine was the worst U.S. mine disaster in nearly 40 years.

The Document Creation Time (t0 in the fragment) forms the right boundary of the Narrative Container, which has, in this genre, a length of approximately a day. The two other time expressions, *Sunday* and *earlier this month* both are Narrative Times for parts of the fragment. With this information now available, it becomes easier to determine what anchor times to relate events to. For example, for *paid*, the event with the identifier e1, we would resolve how it relates to *Sunday*, possibly using included_in as a default. We can then envision to build a temporal graph as in Figure 4. It should be pointed out that the notion of narrative container mentioned here shares some of the features of the notion of "framing" structures, as developed in (Charolles, 2005), as well as the "discourse frame" as used in SDRT.

Note that some of the functionality needed for this is very similar to (i) the heuristics used by BTime to determine what the anchor time is and (ii) the external script that determines what section headers are and how to assign a temporal value to them.

4. Conclusion

We have described the current status of the TARSQI Toolkit, focussing on recent changes and current work. We have not attempted to compare the toolkit with any of the many recent systems that extract temporal expressions, events and temporal relations for English (Chambers and Jurafsky, 2009; Puşcaşu, 2007; Hepple et al., 2007; Bethard and Martin, 2007; Derczynski and Gaizauskas, 2010; Strötgen and Gertz, 2010; Saquete Boro, 2010; Llorens et al., 2010; UzZaman and Allen, 2010; Kolya et al., 2011).

The elements of TTK as described above are in various stages of completion and updates continue to be made. The move to standoff annotation has been wrapped up and BTime is implemented and was inserted into the processing chain. The main remaining issues for the latter are on grammar coverage and speed. Since BTime uses the Early algorithm, it is also sensitive to the pitfalls of Early's $O(n^3)$ time complexity. This can be managed by (i) recognizing and avoiding costly rules that add too many edges t the graph, and (ii) implementing an expected maximum length of a time expression, which will move BTime's time complexity to a linear scale.

Extensions for the medical domain have been implemented, but are often only in existence as external scripts. Some of



Figure 4: Events grouped into their appropriate Narrative Times.

these scripts modify the BLinker module and these modifications need to be more tightly integrated. Finally, design and implementation of the narrative container idea has only recently started.

The TARSQI source code is available on the TimeML website at http://timeml.org/. An updated version, including at least the new version of BTime and the extensions for the medical domain, will be made available in the spring of 2012. This version will also use stand-off annotation throughout and include many other updates like unicode support and greater easy of integration with other annotations.

5. Acknowledgements

This work was supported by the National Institutes of Health, under grant number 5R01NS057484-04. We also like to thank Amber Stubbs, Alex Plotnick and Russell Entrikin, who have all made recent contributions to the code base.

6. References

- James Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843.
- Steven Bethard and James H. Martin. 2007. CU-TMP: Temporal Relation Classification Using Syntactic and Semantic Features. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pages 129–132, Prague, Czech Republic, June. Association for Computational Linguistics.
- Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and their Participants. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pages 602–610, Suntec, Singapore, August. Association for Computational Linguistics.
- M. Charolles. 2005. Framing Adverbials and their Role in Discourse Cohesion from Connection to Forward Labelling. In *Papers of the Symposium on the Exploration and Modelling of Meaning (Sem 05)*, Biarritz.
- Leon Derczynski and Robert Gaizauskas. 2010. USFD2: Annotating Temporal Expressions and TLINKs for TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 337–340, Uppsala, Sweden, July. Association for Computational Linguistics.
- Jay Earley. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102.

- Mark Hepple, Andrea Setzer, and Robert Gaizauskas. 2007. USFD: Preliminary Exploration of Features and Classifiers for the TempEval-2007 Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 438–441, Prague, Czech Republic, June. Association for Computational Linguistics.
- Nancy Ide and Laurent Romary. 2006. Representing Linguistic Corpora and Their Annotations. In *Proceedings* of the Fifth Language Resources and Evaluation Conference (LREC), Genoa, Italy.
- Nancy Ide and Keith Suderman. 2007. GrAF: A Graphbased Format for Linguistic Annotations. In *Proceedings of the Linguistic Annotation Workshop*, pages 1–8, Prague, Czech Republic, June. Association for Computational Linguistics.
- Anup Kumar Kolya, Asif Ekbal, and Sivaji Bandyopadhyay. 2011. A Hybrid Approach for Event Extraction and Event Actor Identification. In Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, pages 592–597, Hissar, Bulgaria, September. RANLP 2011 Organising Committee.
- Hector Llorens, Estela Saquete, and Borja Navarro. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of the* 5th International Workshop on Semantic Evaluation, pages 284–291, Uppsala, Sweden, July. Association for Computational Linguistics.
- Inderjeet Mani and George Wilson. 2000. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 69–76, Hong Kong, October. Association for Computational Linguistics.
- Inderjeet Mani, Ben Wellner, Marc Verhagen, Chong Min Lee, and James Pustejovsky. 2006. Machine Learning of Temporal Relations. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, Sydney. ACL.
- Inderjeet Mani, Ben Wellner, Marc Verhagen, and James Pustejovsky. 2007. Three Approaches to Learning TLINKs in TimeML. Technical Report CS-07-268, Brandeis University, Waltham, United States.
- Georgiana Puşcaşu. 2007. WVALI: Temporal Relation Identification by Syntactico-Semantic Analysis. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 484–487, Prague, Czech Republic, June. Association for Computational Linguistics.
- James Pustejovsky and Amber Stubbs. 2011. Increasing Informativeness in Temporal Annotation. In *Proceed*-

ings of the Linguistic Annotation Workshop V, Portland, Oregon. Association of Computational Linguistics.

- James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *IWCS-5 Fifth International Workshop on Computational Semantics*.
- James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003b. The TimeBank Corpus. *Corpus Linguistics*, pages 647–656, March.
- James Pustejovsky, Robert Knippen, Jessica Littman, and Roser Saurí. 2005. Temporal and Event Information in Natural Language Text. *Language Resources and Evaluation*, 39:123–164, May.
- James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. ISO-TimeML: An International Standard for Semantic Annotation. In *LREC*, Malta, May.
- Estela Saquete Boro. 2010. ID 392:TERSEO + T2T3 Transducer. A systems for Recognizing and Normalizing TIMEX3. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 317–320, Uppsala, Sweden, July. Association for Computational Linguistics.
- Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. Evita: A Robust Event Recognizer For QA Systems. In *Proceedings of the HLT/EMNLP* 2005, pages 700–707.
- Roser Saurí, Marc Verhagen, and James Pustejovsky. 2006a. Annotating and Recognizing Event Modality in Text. In *Proceedings of of the 19th International FLAIRS Conference, FLAIRS 2006*, Melbourne Beach, Florida, USA.
- Roser Saurí, Marc Verhagen, and James Pustejovsky. 2006b. SlinkET: A Partial Modal Parser for Events. In *Proceedings of LREC 2006*, Genoa, Italy.
- Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.
- Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden, July. Association for Computational Linguistics.
- Amber Stubbs and Benjamin Harshfield. 2010. Applying the TARSQI toolkit to augment text mining of EHRs. In *Proceedings of the 2010 Workshop on Biomedical Natural Language (BioNLP 2010)*, July.
- Naushad UzZaman and James Allen. 2010. TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 276– 283, Uppsala, Sweden, July. Association for Computational Linguistics.
- Marc Verhagen and James Pustejovsky. 2008. Temporal Processing with the TARSQI Toolkit. In *Coling 2008:*

Companion volume: Demonstrations, pages 189–192, Manchester, UK, August. Coling 2008 Organizing Committee.

- Marc Verhagen, Inderjeet Mani, Roser Sauri, Jessica Littman, Robert Knippen, Seok Bae Jang, Anna Rumshisky, John Phillips, and James Pustejovsky. 2005. Automating Temporal Annotation with TARSQI. In Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics, Ann Arbor, USA. Demo session.
- Marc Verhagen, Robert Knippen, Inderjeet Mani, and James Pustejovsky. 2006. Annotation of Temporal Relations with Tango. In *Proceedings of LREC 2006*, Genoa, Italy.
- Marc Verhagen. 2005. Temporal closure in an annotation environment. *Language Resources and Evaluation*, 39:211–241, May.
- Marc Verhagen. 2007. Drawing TimeML Relations with TBox. In Graham Katz, James Pustejovsky, and Frank Schilder, editors, *Annotating, Extracting and Reasoning about Time and Events*, volume 4795 of *Lecture Notes in Computer Science*, pages 7–28. Springer.