Temporal Processing with the TARSQI Toolkit

Marc Verhagen
Brandeis University
Computer Science Dept.
Waltham, MA 02454-9110
`marc@cs.brandeis.edu`

James Pustejovsky
Brandeis University
Computer Science Dept.
Waltham, MA 02454-9110
`jamesp@cs.brandeis.edu`

Abstract

We present the TARSQI Toolkit (TTK), a modular system for automatic temporal and event annotation of natural language texts. TTK identifies temporal expressions and events in natural language texts, and parses the document to order events and to anchor them to temporal expressions.

1 Introduction

A keyword-based search is not sufficient to answer temporally loaded questions like "*did Brazil win the soccer world championship in 1970?*" since a boolean keyword search cannot distinguish between those documents where the event *win* is actually anchored to the year 1970 versus those that are not. The TARSQI Project (Temporal Awareness and Reasoning Systems for Question Interpretation) focused on enhancing natural language question answering systems so that temporally-based questions about the events and entities in news articles can be addressed. To explicitly mark the needed temporal relations the project delivered a series of tools for extracting time expressions, events, subordination relations and temporal relations (Verhagen et al., 2005; Mani et al., 2006; Saurí et al., 2005; Saurí et al., 2006a). But although those tools performed reasonably well, they were not integrated in a principled way.

This paper describes the TARSQI Toolkit (TTK), which takes the TARSQI components and integrates them into a temporal parsing framework. The toolkit is different from the system described in (Verhagen et al., 2005) in several major aspects:

1. the components were integrated in a toolkit which, amongst others, split the parsing of properties typical for a particular document type from the temporal parsing of the text

2. a component was added that takes the results from the various components that generate temporal relations and merges them into a consistent temporal graph

3. a new way of visualizing the results was used

In addition, some components were updated and test suites with unit tests and regression tests were added. In this paper, we focus on the merging of temporal links and the visualization of temporal relations.

There has been a fair amount of recent research on extraction of temporal relations, including (Chambers et al., 2007; Lapata and Lascarides, 2006; Bramsen et al., 2006; Bethard and Martin, 2007; Min et al., 2007; Puşcaşu, 2007). However, we are not aware of approaches that integrate temporal relations from various sources in one consistent whole.

All TTK components use the TimeML annotation language (Pustejovsky et al., 2003; Pustejovsky et al., 2005). TimeML is an annotation scheme for markup of events, times, and their temporal relations in news articles. The TimeML scheme flags tensed verbs, adjectives, and nominals with EVENT tags with various attributes, including the class of event, tense, grammatical aspect, polarity (negative or positive), and any modal operators which govern the event being tagged. Time expressions are flagged with TIMEX3 tags, an extension of the ACE 2004 TIMEX2 annotation scheme (tern.mitre.org).

Subordination relations between events, as for example between reporting events and the embedded event reported on, are annotated with the SLINK tag. For temporal relations, TimeML defines a TLINK tag that links tagged events to other events and/or times.

In section 2, we will give a short overview of the toolkit. In section 3, we focus on the component that merges TLINKs, and in section 4 we will dwell on the visualization of temporal relations.

2 Overview of the toolkit

The overall architecture of TTK is illustrated in figure 1 below. 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.
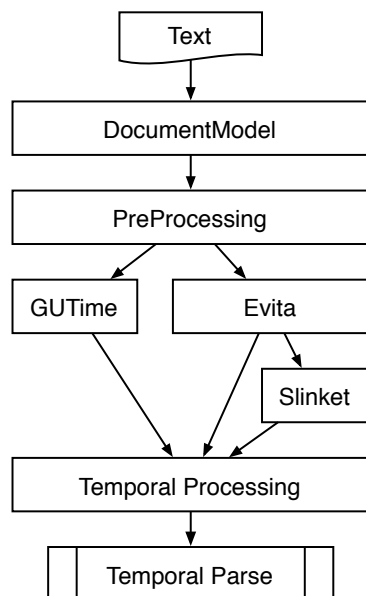

Figure 1: TTK Architecture

The preprocessor uses standard approaches to tokenization, part-of-speech tagging and chunking. GUTime is a temporal expression tagger that recognizes the extents and normalized values of time expressions. 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 is an application developed to automatically introduce SLINKs, which in TimeML specify subordinating relations between pairs of events, and classify them into factive, counterfactive, evidential, negative evidential, and modal, based on the modal force of the subordinating event (Saurí

et al., 2006b). SLINKs are introduced by a well-delimited subgroup of verbal and nominal predicates (such as *regret*, *say*, *promise* and *attempt*), and in most cases clearly signaled by a subordination context. Slinket thus relies on a combination of lexical and syntactic knowledge.

The temporal processing stage includes three modules that generate TLINKs: Blinker, S2T and the TLink Classifier.

Blinker is a rule-based component that applies to certain configurations of events and timexes. It contains rule sets for the following cases: (i) event and timex in the same noun phrase, (ii) events and the document creation time, (iii) events with their syntactically subordinated events, (iv) events in conjunctions, (v) two main events in consecutive sentences, and (vi) timexes with other timexes. Each of these rule sets has a different flavor. For example, the rules in (vi) simply calculate differences in the normalized ISO value of the timex tag while the rules in (v) refer to the tense and aspect values of the two events. Blinker is a re-implementation and extension of GutenLink (Verhagen et al., 2005).

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.

The TLink Classifier is a MaxEnt classifier that identifies temporal relations between identified events in text. The classifier accepts its input for each pair of events under consideration as a set of features. It is trained on the TimeBank corpus (see www.timeml.org).

Of the three TLINK generating components, S2T derives a relatively small number of TLINKs, but Blinker and the classifier are quite prolific. In many cases the TLINKs derived by Blinker and the classifier are inconsistent with each other. The system in (Verhagen et al., 2005) used a simple voting mechanism that favors TLINKs from components that exhibit higher precision. In addition, if confidence measures are available then these can be used by the voting mechanism. However, this approach does not factor in consistency of temporal relations: choosing the TLINKs with the highest probability may result in TLINKs that are inconsistent. For example, say we have two TLINKs: BEFORE(x,y) and BEFORE(y,z). And say we have

two competing TLINKs, derived by Blinker and the classifier respectively: BEFORE(x,z) and BEFORE(z,x). If the second of these two has a higher confidence, then we will end up with an inconsistent annotation. In the following section we describe how in TTK this problem is avoided.

3 Link Merger

The link merger, together with the three TLINK-generating components, is part of the temporal processing module of TTK, as shown in the diagram in figure 2 below.
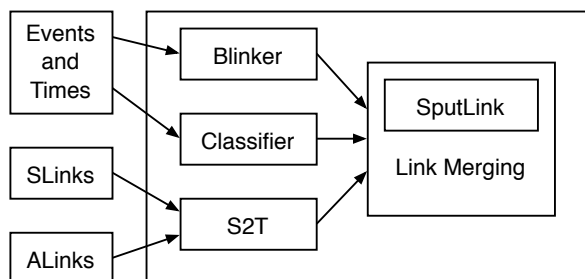

Figure 2: TTK Temporal Processing

The link merging component uses a greedy algorithm to merge TLinks into a consistent whole. First all links are ordered on their confidence score. 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 that has no constraints expressed on the edges. Those constraints are added by the temporal links. Links are ordered on confidence score and are added one by one. Each time a link is added a constraint propagation component named Sputlink, based on Allen's interval algebra (Allen, 1983; Verhagen, 2005), 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 prefered over lower precision links.

4 Visualization

Providing a good visualization of a temporal graph can be tricky. A table of temporal relations is only useful for relations inside sentences. Full graphs, like the ones generated by GraphViz (http://www.graphviz.org/), do not make it that much easier for the reader to quickly obtain a picture of the temporal structure of the document. Timelines can be misleading because so many events in a document cannot be ordered with respect to a time stamp.

TTK uses a visualization scheme named TBox (Verhagen, 2007). It uses left-to-right arrows, box inclusion and stacking to encode temporal precedence, inclusion, and simultaneity respectively (see figure 3).
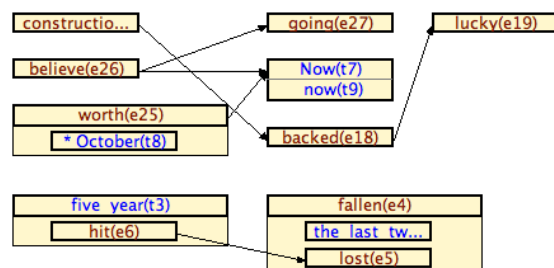

Figure 3: The TBox Representation

This visualization makes it easier to convey the temporal content of a document since temporal relations are strictly 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.

5 Conclusion and Future Work

We have described TTK, a toolkit that integrates several components that generate tags to mark up events and time expressions, as well as non-consuming tags that encode relations between events and times. TTK includes a module that combines potentially conflicting temporal relations into a consistent temporal graph of a document, which can be succinctly displayed using the TBox representation.

In current work, we are exploring how to split up the task of temporal relation extraction into more subtasks and write specialized components, both rule-based and machine learning based, to extract temporal relations for that task. The link merging would then have many more input streams, each with their own reported reliability.

The TARSQI Toolkit can be downloaded from http://timeml.org/site/tarsqi/toolkit/.

191

Acknowledgments

References

Allen, James. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843.

Bethard, Steven 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.

Bramsen, Philip, Pawan Deshpande, Yoong Keok Lee-and, and Regina Barzilay. 2006. Finding temporal order in discharge summaries. In *Proceedings of EMNLP*.

Chambers, Nathanael, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 173–176, Prague, Czech Republic, June. Association for Computational Linguistics.

Lapata, Mirella and Alex Lascarides. 2006. Learning sentence-internal temporal relations. *Journal of Artificial Intelligence Research*, 27:85–117.

Mani, Inderjeet, 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.

Min, Congmin, Munirathnam Srikanth, and Abraham Fowler. 2007. LCC-TE: A hybrid approach to temporal relation identification in news text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 219–222, Prague, Czech Republic, June. Association for Computational Linguistics.

Puşcaşu, Georgiana. 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.

Pustejovsky, James, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *IWCS-5 Fifth International Workshop on Computational Semantics*.

Pustejovsky, James, Robert Knippen, Jessica Littman, and Roser Saurí. 2005. Temporal and event information in natural language text. *Language Resources and Evaluation*, 39:123–164, May.

Saurí, Roser, 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.

Saurí, Roser, 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.

Saurí, Roser, Marc Verhagen, and James Pustejovsky. 2006b. SlinkET: A partial modal parser for events. In *Proceedings of LREC 2006*, Genoa, Italy.

Verhagen, Marc, 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.

Verhagen, Marc. 2005. Temporal closure in an annotation environment. *Language Resources and Evaluation*, 39:211–241, May.

Verhagen, Marc. 2007. Drawing TimeML relations with TBox. In Katz, Graham, 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.