Creating and retrieving tense and aspect annotations with GraphAnno, a lightweight tool for multi-level annotation

Volker Gast
Friedrich Schiller University Jena
volker.gast@uni-jena.de

Lennart Bierkandt
Friedrich Schiller University Jena
post@lennartbierkandt.de

Christoph Rzymski
Friedrich Schiller University Jena
christoph.rzymski@uni-jena.de

1 Introduction

In this paper, we propose an annotation scheme for the manual annotation of tense and aspect in natural language corpora, as well as an implementation using GraphAnno, a configurable tool for manual multi-level annotation. The annotation scheme is based on Klein's (1994) theory of tense and aspect, arguably the most widely accepted theory in this domain (cf. also Klein and Li 2009). One of the most important features of Klein's theory is that in addition to the time span during which a situation obtains (the 'time of situation'/TSit), it makes use of the concept of 'Topic Time' (TT), which is related to, but different from, Reichenbach's (1947) reference point 'R' (cf. Derczynski and Gaizauskas 2013). Given that the resulting annotations cannot be mapped one-to-one to words or constituents, and as they are partially retrieved from the context, a semantic layer of annotation is needed, in addition to the structural one. The multi-level approach advocated here also allows us to annotate temporal relations across sentences.

Section 2 provides some background on GraphAnno. Some ontological prerequisites of the annotation of tense and aspect are established in Section 3, and the concept of 'Topic Time' is introduced. Sections 4 and 5 demonstrate the annotation of temporal relations within and beyond the sentence, respectively. Section 6 describes the query language of GraphAnno, and Section 7 contains an outlook.

2 Some background on GraphAnno

GraphAnno was originally designed as a prototype for a more powerful annotation tool, Atomic (cf. Druskat et al. 2014), in a project on multi-level annotation of cross-linguistic data.[1] The tool has been used in various corpus-based projects (e.g. Gast 2015), and it has proven a stable and user-friendly application. Moreover, GraphAnno has some functions that Atomic lacks, specifically for searching and filtering (cf. Sect. 6 and Gast et al. 2015). It was therefore published in 2014, and will continue to be maintained.[2]

GraphAnno is so called because the corpus data is program-internally represented, and also visually displayed, as a graph, consisting of annotated nodes and edges. The application is platform-independent, but it requires Graphviz[3] and Ruby.[4] It handles dependencies on other libraries using the RubyGems package manager. An exe-file for easy use on a Windows system is available, bundling the required Ruby runtime environment. The tool has a browser-based interface and is operated via a command line at the bottom of the browser window. Annotations are created with one-letter commands such as `n` (create a node), `g` (grouping nodes into constituents), `e` (create an edge), `d` (deleting nodes or edges) and `a` (annotation of nodes and edges with attribute-value pairs), followed by their arguments. Navigation

[1] LinkType, sponsored by the German Science Foundation (DFG, grant GA-1288/5). Financial support from this institution is gratefully acknowledges; see also http://www.linktype.iaa.uni-jena.de. [2] https://github.com/LBierkandt/graph-anno
[3] http://www.graphviz.org [4] http://www.ruby-lang.org

and additional functions such as filtering, searching and configurations are accessed and controlled with key bindings and function keys.

GraphAnno has an import function, and some preprocessing functionalities are implemented, e.g. punkt segmenters. It uses JSON-files for native storing. Scripts and converters are available or under construction for other corpora, e.g. the BioScope corpus (Vincze et al., 2008) and Timebank 1.2,[5] and for corpus formats like those accessible through NLTK[6] modules.

3 The elements of time and tense annotation

We adopt Klein's (1994: Ch.4) 'Basic Time Concept'. Points in time are identified with real numbers ($r \in \mathbb{R}$), time spans are intervals (i = [r_i,r_2]). Relations between intervals, e.g. of anteriority, can be established by relating the temporal atoms of a time span to each other, for instance:

(1) i_1 is ANT(ERIOR) to i_2 iff:
$\forall a \in i_1, \forall b \in i_2$: $a < b$

The analysis and annotation of time and tense requires establishing a system of types of intervals and relations between such intervals. The first formal system of tense logic was proposed by Reichenbach (1947). Reichenbach (1947) uses three points in time, 'S' (the moment of speech), 'E' (the event) and 'R' (a reference point). S and E are obviously indispensable components of any theory of tense and aspect and are, more or less directly, also part of prominent annotation schemes such as the (ISO-)TimeML language (Pustejovsky et al., 2005; Schilder et al., 2007). However, TimeML does not provide for a reference point. This is one of the reasons why "[i]n many ways, TimeML's tense system is less expressive than that of Reichenbach's" (Derczynski and Gaizauskas, 2013, 6).

As Derczynski and Gaizauskas (2013, 1) note, many efforts are currently being made to improve "reference point management" in computational linguistics. While we believe that this is a promising and in fact necessary development, we refer to a more recent formal approach to tense and aspect. Reichenbach's system is known to exhibit some weaknesses, as was already pointed by Comrie (1981), among others, who published a monograph with a theory of his own a few years later (Comrie 1985; cf. also Declerck 1986). The most comprehensive theory of tense in a Reichenbachian tradition so far has been proposed by Klein (1994), and we think it is fair to say that in theoretical linguistics, Klein (1994) is regarded as a standard in this domain. As we see no reason to refer to the older and, in many ways, fragmentary system of Reichenbach (1947), while a more comprehensive and 'cleaner' follow-up theory is available in the form of Klein (1994), we refer to the latter theory in our proposal.

3.1 Klein's (1994) Topic Time

Like Reichenbach (1947), Klein (1994) uses three prime elements in his theory, which he calls 'time of utterance'/TU (≈ Reichenbach's 'S'), 'time of situation'/TSit (≈ 'E'), and ' Topic Time'/TT. Klein's Topic Time is similar, but not identical, to Reichenbach's reference point R (e.g. insofar as it can be an interval). It is "the time span to which a speaker's claim is confined" (Klein, 1994, 6). Let us consider an example for illustration. In (2) speaker A asks a question about a specific point in time, 6am yesterday, which is established as a Topic Time. Speaker B provides information about this Topic Time.

(2) A: What was the weather like at 6pm yesterday?
B: It was raining.

Example (2) already shows why we need multi-level annotations to capture a Reichenbach/Klein-style tense semantics: There is no structural constituent corresponding to the Topic Time in B's answer.

One of the most important distinctions that can be made using Klein's Topic Time is the one between the Simple Past and the Present Perfect in English. According to Klein (1994), the Simple Past is used

[5] http://www.timeml.org/site/timebank/timebank.html [6] https://www.nltk.org/

when TT is located before TU/t_0 (TT $<$ t_0). The Present Perfect, by contrast, is used when TT includes the moment of utterance (TT $\supseteq$ t_0). Consider the examples in (3).

(3) a. I have lived in New York.
b. I lived in New York.

Both (3a) and (3b) say that there is a situation of the type 'living in New York' in which the speaker participated, and which is located before t_0. The difference concerns the time spans about which information is provided. (3a) provides information about t_0. It could be paraphrased as '*It is now the case that* I have lived in New York', and a likely implicature is that 'I (*now*) know (what it is like to live in) New York'. (3b) makes a statement about a time span in the past. A likely context for (3b) would be a question like 'What did you do in 1987'? Note that the semantic difference between the Present Perfect and the Simple Past is reflected in the fact that the Present Perfect is only compatible with temporal adverbials denoting time spans which contain t_0, while the Simple Past only allows time spans that precede t_0.

Being an important component of tense logic in general, the distinction between TT and TSit has a number of further advantages in the context of text annotation. Narratives are organized around TTs, as has also been noticed by Derczynski and Gaizauskas (2013, 4) for Reichenbach's R: "Observations during the course of this work suggest that the reference time from one sentence will roll over to the next sentence, until it is repositioned explicitly by a tensed verb or time". The location of the Topic Time depends on the tense used. In the case of the Simple Past, TTs are lined up sequentially. The Progressive aspect, by contrast, does not have any such effect of 'TT advancement'. Consider the examples in (4).

(4) a. At 3pm, John sat on the chair, looked at his watch and sang a song.
b. At 3pm, John was sitting on the chair, looking at his watch and singing a song.

In (4a), the events happen sequentially, and the Topic Times form a chain. (As a consequence, the situations are also in temporal sequence, being related to the Topic Time through the [perfective] aspect in each case.) In (4b), there is only one Topic Time, 3pm, and all events described in the sentence 'surround' it. This type tense configuration – TT being fully included in TSit – is exactly what characterizes the progressive aspect, according to Klein (1994). What the examples in (4) show is that the Topic Time needs to be specified for each event, and that it cannot be recovered on purely structural grounds.

3.2 The grammatical categories of tense and aspect

According to Klein (1994), tense is a relation between TU/t_0 and TT. The type of predication expressed by the grammatical category of tense thus takes the form shown in (5). The category 'Past' is regarded as a (morphological) feature, interpreted as a one-place predicate.

(5) $[\![\text{PAST}]\!] = \lambda i\ [i\ \text{ANT}\ t_0]$

ASPECT expresses a relation between the TT and TSit. In (2) above, speaker B expresses that the Topic Time (6am) is fully included in TSit (the situation of raining). The denotation of the aspectual category 'progressive' can thus be represented as shown in (6).

(6) $[\![\text{PROG}]\!] = \lambda i \lambda s[i \subset s]$

4 Annotating structure and function

We will assume that, in English, tense and aspect are structurally represented in the form of features or, more precisely, attribute-value pairs. The VP *was sleeping* can be represented as shown in Figure 1.

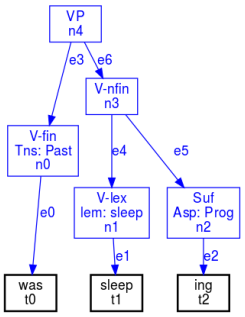

Figure 1: Structure

GraphAnno offers users the possibility to define a theoretically infinite number of levels of annotation, but we can work with the two

levels that are preconfigured, a structural one (s-layer) and a functional one (f-layer). The tree in Figure 1 belongs to the structural layer. Annotations relating to tense and aspect can now be added on the f-layer. Denotations of nodes will be indicated according to the following conventions:

- Relations between intervals are indicated by capitalized abbreviations like 'ANT' (for anteriority).
- Predicates are separated from their arguments by a dash, the arguments are separated by a comma, e.g. 'IN-i,s' for 'i is included in s'.
- λ-bound variables are written between brackets, e.g. '[i]' for a λ-bound variable $\lambda i[\dots i \dots]$. The unsaturated predicate $\lambda i \lambda s[\textsc{ant}(i)(s)]$ is thus represented as 'ANT-[i],[s]'.

In Figure 2, the s-layer is blue, the f-layer green. The nodes for the finite verb, the lexical verb and the progressive aspect marker are each linked to a node on the f-layer (Tns, Sit, Asp). The edges linking the functional nodes to the structural ones are of category 'dn', standing for 'denotation'. The [Past]-feature of the finite verb is interpreted as 'dn:[i]<t0'. The lexical predicate *sleep* denotes a situation (Sit) of sleeping. The [Prog]-feature (corresponding to the *ing*-suffix) denotes the progressive aspect, which, in accordance with Klein (1994), indicates that the Topic Time, TT, is fully included in the time of the situation, TSit. This is here represented as 'IN-[i],[s]' in the Asp-node.

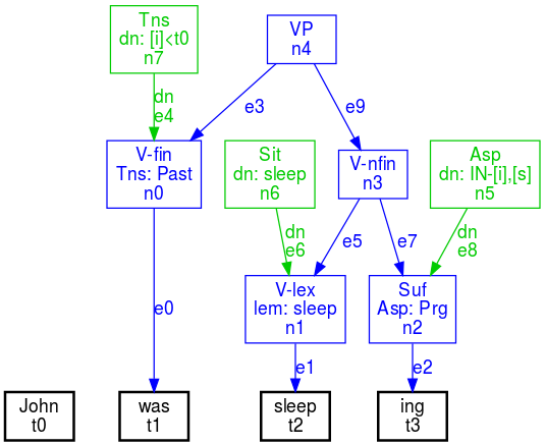

Figure 2: Nodes and their denotations

The temporal and aspectual predications can now be linked to their arguments. Every finite predicate is associated with a Topic Time. The (exact) Topic Time is mostly implicit (cf. below), but it can be made explicit with a temporal adverbial like *at six*. This adverbial denotes a (minimal) time span i, a point in time. We represent points in time in the format 'day/hour/minute/second'. A plus or minus sign indicates time specification relative to t_0. Accordingly, '-1/6/30/0' stands for 'one day before t_0 at 6:30 am'. Time nodes carry an s-attribut for the start and an e-attribute for the end of a time span. In the case of a point in time, the s- and e-attributes are identical. Figure 3 shows the graph in which the tense and aspect predications are linked to their arguments (some structural annotations are omitted for better visibility).[7] The relevant edges are labelled 'arg1' and 'arg2'.

Let us consider more complex cases like the ones in (7) (suggested to us by a reviewer):

(7) a. John taught three hours every week last semester.

b. John has been teaching at Oxford since 2009.

The Topic Time of (7a) is specified as 'every week last semester'. It can be interpreted as a generalized quantifier. As TT takes wide scope, it can be represented as shown in (8). The choice of tense (Simple Past) is in accordance with the fact that TT is located before t_0, and the perfective/non-progressive aspect is used because each instance of teaching (TSit) is fully included in each instance of w (TT).

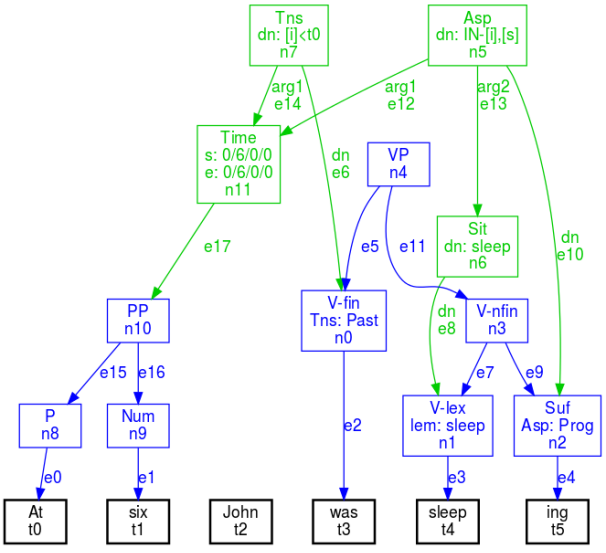

Figure 3: Nodes and relations between them

(8) $\forall w \subseteq [\![\text{last semester}]\!]$: John taught three hours in w

[7] Note that GraphAnno allows users to filter and hide elements with specific properties (such as membership to a given level) for better visibility; cf. Gast et al. (2015) for more information and illustration.

(7b) represents a well-known problem of English temporal semantics, the Present Perfect Progressive. We assume that this tense is interpreted in an additive manner, combining the meaning of the Perfect aspect (anteriority) with that of the Progressive aspect (inclusion of TT in TSit). It says that a sub-event of s, s', which is of the same type as s (teaching [someone]), is located before TT, and that TT is fully included in TSit. (7b) is thus interpreted as shown in (9). For the annotation graph, this means that two Asp-nodes will be linked to the same (Topic) Time node.

(9) for TT $= t_0$:
$\exists y \exists s \exists s'_{\subseteq s}$:
$s = [2009, t_0] \wedge \text{TEACH}(y)(John)(s) \wedge \text{TEACH}(y)(John)(s') \wedge \underset{\text{progressive}}{\text{TT} \subseteq s} \wedge \underset{\text{perfect}}{s' \leq \text{TT}}$

5 Beyond the sentence

We now have a framework for the annotation of tense and aspect within the sentence. We want to be able to annnotate (and retrieve) temporal relations across sentences as well. In order to be able to annotate contextual, often implicit temporal information, we add 'context tokens', represented by a hash, at the beginning of each sentence. They contain information about the Topic Time. Implicit Topic Times are often identical to the preceding sentence, or they correspond to an immediately following time span.

Figure 4 shows an example (*He came at six. The sun had sunk*). Again, some structural annotations are omitted. The context node, here tokenized as `t5`, carries an annotation at the functional level which provides the Topic Time for the second sentence. It is copied from the first sentence. In this way text-level temporal structures can be annotated and, as we will see in the next section, retrieved.

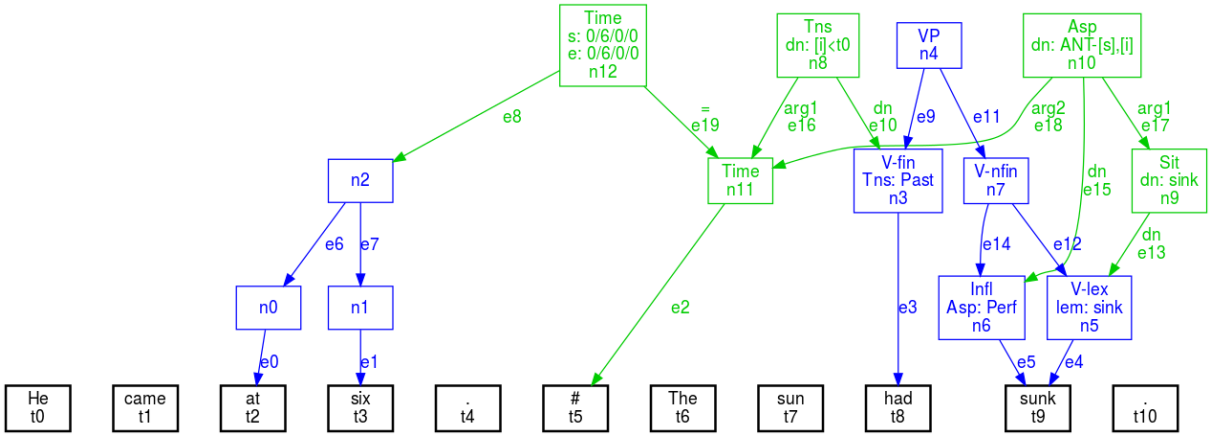

Figure 4: Annotations beyond the sentence boundary

6 Retrieving temporal configurations

GraphAnno also has a powerful yet transparent query language. The user specifies a graph fragment by describing it in terms of attribute-value pairs associated with nodes, as well as edges between nodes. The following query retrieves temporal configurations of the type shown in Figure 4.

(10)
```
node @a cat:Tns & dn:[i]<0          # define Past tense node
node @b cat:Asp & dn:ANT-[s],[i]    # define Perfect aspect node
node @c cat:Time                    # define Time node
edge @a@c                           # edge between @a and @c
edge @b@c                           # edge betweeb @a and @b
```

Any graph fragment matching the query is highlighted visually and can be exported into a data frame.

7 Outlook

We have focused on the manual annotation of tense and aspect configurations in English. The annotation scheme can be regarded as an implementation of Klein's (1994) theory of tense and aspect and is thus, as we believe, fully interpretable linguistically speaking. We hope to have shown that GraphAnno's unrestrictive approach to annotation allows for the implementation and subsequent testing of linguistic theories, without being specifically tailored to any specific theory.

Some of the semantic annotations used for illustration are obviously redundant (since predictable from structural ones) and can largely be automated. For instance, the denotations of morphological features like [Past] and [Prog] are largely (though not entirely) invariant. A more challenging task consists in figuring out the relationships between Topic Times across sentences. Annotation experiments will show to what extent such text-level annotations are amenable to meachine learning.

References

Comrie, B. (1981). On Reichenbach's approach to tense. In R. A. Hendrick, C. S. Masek, and M. F. Miller (Eds.), *Proceedings of the 7th meeting of the Chicago Linguistics Society*, pp. 24–30.

Comrie, B. (1985). *Tense*. Cambridge: Cambridge University Press.

Declerck, R. (1986). From Reichenbach (1947) to Comrie (1985) and beyond: Towards a theory of tense. *Lingua 70*, 305–364.

Derczynski, L. and R. Gaizauskas (2013). Empirical validation of Reichenbach's tense framework. In *Proceedings of the 10th International Conference on Computational Semantics*.

Druskat, S., L. Bierkandt, V. Gast, C. Rzymski, and F. Zipser (2014). Atomic: An open-source software platform for multi-level corpus annotation. In J. Ruppert and G. Faaß (Eds.), *Proceedings of the 12th Konferenz zur Verarbeitung natrlicher Sprache (KONVENS 2014), October 2014*, pp. 228–234.

Gast, V. (2015). On the use of translation corpora in contrastive linguistics: A case study of impersonalization in english and german. *Languages in Contrast 15*(1), 4–33.

Gast, V., L. Bierkandt, and C. Rzymski (2015). Annotating modals with GraphAnno, a configurable lightweight tool for multi-level annotation. In *Proceedings of the Workshop on Models for Modality Annotation, held in conjunction with IWCS 11, 2015*.

Klein, W. (1994). *Time in Language*. London: Routledge.

Klein, W. and P. Li (Eds.) (2009). *The Expression of Time*. Berlin: de Gruyter Mouton.

Pustejovsky, J., R. Ingria, R. Saurí, J. Castaño, J. Littman, R. Gaizauskas, and A. Setzer (2005). The specification language TimeML. In I. Mani, J. Pustejovsky, and R. Gaizauskas (Eds.), *The Language of Time: A Reader*. Oxford: Oxford University Press.

Reichenbach, H. (1947). *Elements of Symbolic Logic*. New York: Macmillan & Co.

Schilder, F., G. Katz, and J. Pustejovsky (Eds.) (2007). *Annotating, Extracting and Reasoning about Time and Events*. Heidelberg: Springer.

Vincze, V., G. Szarvas, R. Farkas, G. Móra, and J. Csirik (2008). The BioScope corpus: Biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics 9*(S-11).