

Time after Time: Representing Time in Literary Texts

Greg Lessard¹ and Michael Levison²

¹ French Studies, Queen's University, Canada
greg.lessard@queensu.ca

² School of Computing, Queen's University, Canada
levison@cs.queensu.ca

Abstract. The representation of temporal information in text represents a significant computational challenge. The problem is particularly acute in the case of literary texts, which tend to be massively underspecified, often relying on a network of semantic relations to establish times and timings. This paper shows how a model based on threaded directed acyclic graphs makes it possible to capture a range of subtle temporal information in this type of text and argues for an onomasiological approach which places meaning before form.

Keywords: literary narrative, representation of time, directed acyclic graph, DAG

1 Time and Text

This paper deals with the representation of temporal phenomena in literary texts. As a result, it builds on work from a number of fields.³ Thus, it takes as given the longstanding observation from philosophy that time is not a simple issue of days, months and years, but reflects issues of perception and culture [1]. At the same time, it assumes that the study of temporal phenomena will be enhanced by use of a formal representation [2]. It further assumes the traditional narratological distinction between the information which underlies a text, variously known as the *fabula* or *histoire* and which we will henceforth call the **story** and some particular instantiation of this in text, often called the *sjuzhet* or *récit*, which we will henceforth call the **narrative** [3].

The analysis of temporal phenomena in text has been pursued from several perspectives. Chief among them is the tagging of texts in order to capture their temporal elements by means of markup schemes like TimeML [4], in order to permit querying and reasoning on the basis of this markup. Since such schemes begin from the text itself, they focus on phenomena in the *narrative* rather than in the *story*. Work in narrative generation [5], or on dynamically generated narratives [6], on the other hand, often begins with a semantic representation of the *story* and seeks to instantiate it in generated *narrative*.

³ In what follows, for lack of space, we restrict ourselves to citing some typical examples from a vast literature.

To put this another way, most work on existing texts adopts a *semasiological* perspective (from form to meaning), while work on generation adopts an *onomasiological* perspective (from meaning to form) [7]. Interestingly, we find fewer attempts to apply an onomasiological approach to existing literary texts: in other words, the attempt to represent formally the *story* behind the existing narrative. This is no doubt due in part to the fact that such an enterprise requires human intelligence to capture the complex interplay of linguistic, encyclopedic and cultural knowledge involved. One of the more developed attempts in this area is no doubt work by Elson [8], which proposes the concept of the *Story Intention Graph* or SIG, which captures the surface text, the timelines, and the reader’s evaluation of the goals, beliefs and plans of characters. However, despite its interest, the SIG model does not entirely achieve the level of onomasiological abstraction we are seeking. In what follows, we will attempt to illustrate this by examining a number of time-related phenomena in several literary texts.

2 DAGs and Threading

The story which underlies a literary text may be represented by means of a **directed acyclic graph**, henceforth DAG, composed of nodes connected by unidirectional edges. The acyclicity requires that no sequence of edges may return to an earlier node. The nodes carry segments of meaning represented by **semantic expressions**. These are functional representations, described in [9]. Each semantic expression is composed of elements drawn from a semantic lexicon.

A simple example might be `stab(brutus, caesar)`, where the two entities `brutus` and `caesar` denote certain human beings, and `stab` is a function which takes two entities and returns a completion⁴ having the obvious meaning. On the basis of functions such as these, it is possible to construct the DAG shown in Fig. 1.⁵

The unidirectional edges between the various nodes represent semantic **dependency**, that is, the fact that subsequent nodes depend upon information contained in previous nodes, and by transitive closure, parents of previous nodes. So, in Fig. 1, the expression `stab(brutus, caesar)` depends on the fact that Caesar is at the Senate, while Caesar being dead depends on the stabbing. The relation carried by edges may be one of order (one node occurs before another), or of some sort of causality, whereby a subsequent node is made possible by a previous node. In addition, nodes which convey a coarse level of meaning may themselves be refined into DAGs at a finer level. And so on, recursively.

⁴ A completion may be thought of as the semantic equivalent of an utterance, an entity as the semantic equivalent of a noun, and an action as the semantic equivalent of a verb.

⁵ The DAGs shown here were constructed with yEd (http://www.yworks.com/en/products_yed_about.html), which generates a GraphML representation for each graph. For simplicity, we have ignored representation of tense and aspect in these examples, although the formalism permits this.

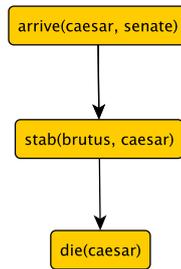


Fig. 1. A DAG for the various states of Caesar

Since a DAG exists prior to any text which represents it, a text may begin at the start of the DAG and follow the nodes, as in *Caesar arrived at the Senate, then Brutus stabbed him, then he died*, or alternatively at the end, as in *Caesar died because Brutus stabbed him after his arrival at the Senate*, in the middle, as in *Brutus stabbed Caesar after he arrived at the Senate, and then he died*, or even in a zigzag order, as in *Caesar arrived at the Senate and then died because Brutus stabbed him*.⁶

Each of these narrations may be represented by a sequence of nodes, in other words, a thread, showing the order in which the meaning carried by the nodes is presented to the reader. Note that the thread passes through some or all of the nodes, but need not follow the edges of the DAG. Nor is it constrained to be acyclic: it may visit the same node more than once. An example of this is provided by a narration in which the same event is recounted twice. To take an extreme case, in the movie *Groundhog Day* (<http://www.imdb.com/title/tt0107048/>), the character Phil relives the same day and its events many times.

In our DAGs, we represent threads by a dotted line to distinguish them from the edges of the DAG. By threading coarse or refined DAGs, the narration can be at different levels of detail. In addition, a single DAG may be traversed by multiple threads representing, among other things, different points of view. So, for example, suppose that a third party, say Livia, finds Caesar's dead body, observes the stab wounds, and concludes that a previously living Caesar has been killed. From the point of view of the Livia thread, the 'Caesar is dead' node is traversed before the stabbing node (although from Livia's point of view, it may not be clear who has done the stabbing). Alternatively, a fourth character may observe a stabbing in the distance, then on approach note that the stabbee is Caesar, assumed alive until that point, and finally learn that Caesar is dead.

3 Relative and Absolute Timestamps

Within the DAG model, the simple chronological ordering of events or activities requires no extra features except perhaps 'colouring' certain edges to distin-

⁶ For more examples, see [10].

guish between those which denote chronological dependence and those whose dependence is based on other reasons. Figure 1 above illustrates this. However, more complex temporal relationships such as ‘while’ can be signified by nodes indicating relative or absolute times, as in:

```

reftime(7)                {relative time}
exacttime(0900, 5, Mar, 2013) {exact time}

```

Consider, for example, the DAG shown in Fig. 2. Here, both `event1` and `event2` take place after `reftime(7)` and before `reftime(8)`.⁷ If no other activities take place in the same context, we might conclude that while `event1` was taking place, `event2` was happening elsewhere. Both events conclude before `event4`. In addition, `event3` occurs after `event1`, but it may have started before or after `reftime(8)`; and there is no information about its relation to `event4`. Additional arguments can be added to specify whether an event is durative or punctual, because nothing says that `event1` actually began at `reftime(7)` and ended at `reftime(8)`. The function `exacttime()` allows us to anchor parts of the DAG at, or more precisely after, specific moments.

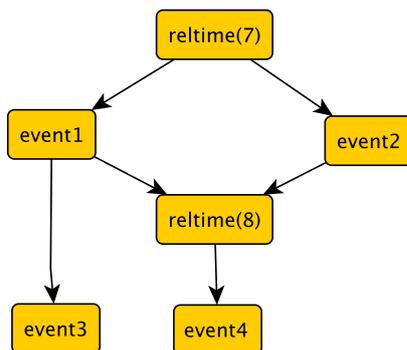


Fig. 2. A DAG showing relative times and events

4 Some Empirical Tests of the Formalism

To empirically test the model proposed here, we will examine several actual texts. Of course, these represent only a small selection of a vast range of temporal phenomena. Our object is simply to show how the proposed model may be applied.

⁷ The parameters to `reftime`, by the way, are irrelevant; they are included only for the purposes of this commentary.

4.1 Prolepsis

As noted earlier, a literary text may bring into play a variety of perspectives. One of these is **prolepsis**, or foreknowledge of a future event. Consider the following example from Homer's *Iliad*.⁸ Achilles asks Zeus for success in a battle and that Patroclus survive the battle. Zeus grants the first wish, but not the second.⁹ As a result, he (Zeus) and by extension, we, as readers, know that Patroclus will die. However Patroclus himself is unaware of this. We may represent this part of the story by means of the DAG shown in Fig. 3, which contains two sets of dependencies, one which links Zeus to the decision that Patroclus will die, and the other which links Patroclus to his fighting and dying. We may then capture the temporality of the narrative by threading this DAG.¹⁰

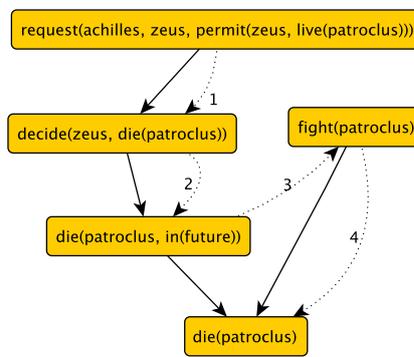


Fig. 3. A DAG for part of the *Iliad*

An example like this may seem arcane, but cases of multiple points of view, or multiple threading, are found in a variety of textual models. Thus, in a murder mystery, the detective comes to understand the ordering of particular events, including the murder, and may subsequently explain this to the reader.

4.2 Parallel Series of Events

Consider the following passage from the Conan Doyle story entitled *The Red-headed League*.¹¹

⁸ This instance of prolepsis is discussed in [11].

⁹ *Iliad*, ch. 16, v. 249, <http://classics.mit.edu/Homer/iliad.html>.

¹⁰ Threads are shown here by numbered dotted lines to indicate their order, while dependencies are shown by unnumbered solid lines.

¹¹ First published in the *Strand* magazine in 1891. See <http://www.gutenberg.org/ebooks/1661>.

When I saw him that afternoon so enwrapped in the music at St. James's Hall . . .¹²

“You want to go home, no doubt, Doctor,” he remarked as we emerged.

“Yes, it would be as well.”

“And I have some business to do which will take some hours. . . . today being Saturday rather complicates matters. I shall want your help to-night.”

“At what time?”

“Ten will be early enough.”

“I shall be at Baker Street at ten.”

. . . It was a quarter-past nine when I started from home and made my way . . . to Baker Street. . . . On entering his room I found Holmes in animated conversation with two men, . . .

The text itself provides two absolute times, one prescriptive, that of the time when Watson is to meet Holmes, and the other descriptive, the time reported by Watson for his leaving home. Another more approximate time is also provided, the fact that Watson and Holmes are listening to music in St James's Hall on a Saturday afternoon. All of these could be marked up in the text itself. However, others would provide a greater challenge. On Watson's return to meet Holmes, he discovers that others are present, presumably at Holmes' invitation. The chronology of Watson's activities is provided only by its placement in the text, between the conversation with Holmes and the return to meet Holmes, while the arrival of the others cannot be marked up at all at the textual level since it is not even mentioned in the text. Such a model provides a serious challenge to a semasiological markup, for obvious reasons. However, it may be easily represented by a DAG, as shown in Fig. 4.

Note that the nodes of the DAG are all enclosed in a higher-level node situated on Saturday. This ‘envelope’ provides the framework for the detailed events. However, within this envelope, a branching occurs, separating Watson's explicitly noted activities from those which we must suppose Holmes to have accomplished. The two series are bracketed between a relative temporal marker (the moment when Watson and Holmes leave each other) and an absolute temporal marker (Watson's arrival at Holmes' lodgings around 10).

4.3 Access to Encyclopaedic Information

Reading a text is not a simple activity. Among other things, it requires a constant reference to background ‘encyclopaedic’ information. The nature of this information will vary from reader to reader. As an illustration, consider the following paragraph, which opens Flaubert's novel *Salammbô*.¹³

¹² Several non-pertinent elements of the text have been elided. These are shown by suspension points.

¹³ We provide here the English translation from <http://www.gutenberg.org/files/1290/1290-h/1290-h.htm#link2HCH0001>.

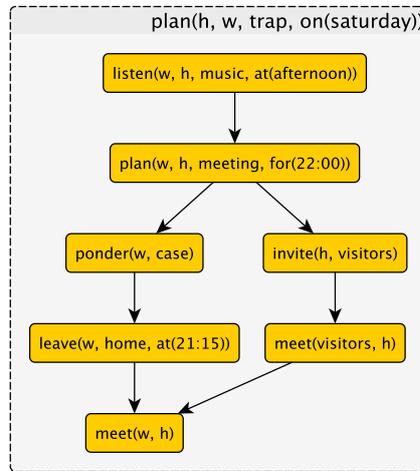


Fig. 4. A DAG for part of the *Red-headed League*

It was at Megara, a suburb of Carthage, in the gardens of Hamilcar. The soldiers whom he had commanded in Sicily were having a great feast to celebrate the anniversary of the battle of Eryx, and as the master was away, and they were numerous, they ate and drank with perfect freedom.

At the most basic level, any reader may use the tense (*had commanded*) and some lexical items (*anniversary*) to determine the anteriority of the battle of Eryx with respect to the feast. However, more educated readers will probably be able to use the proper name *Carthage* to locate the text in the far past, while even more educated readers will be able to use the names *Hamilcar* and *Eryx* to place the feast after the period 244-242 BCE.

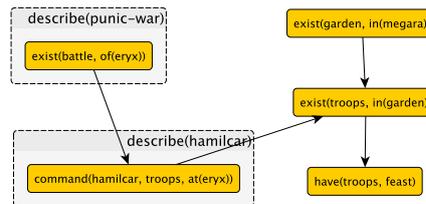


Fig. 5. A DAG for the introduction to *Salammbô*

We may represent the interplay between what is given by the text and the information available to the reader (which, importantly, is also representable by a DAG) as shown in Fig. 5, where we see that the node `exist(troops...)`,

represented in the text, depends on the node `command(hamilcar...)` also represented in the text. However, this latter node is a subnode of the higher-level node `describe(hamilcar)`, which provides information (including temporal information) not present in the text. Similarly, the node `exist(battle...)`, present in the text, is part of another higher-level node (`describe(punic-war)`), which contains more detailed encyclopaedic information.

This model captures both the temporal elasticity provided by the interplay of logical dependency and the varying levels of temporal assignment noted above. To put this another way, it captures the set of readings which the same text may carry for different readers. In particular, different readings may thread this DAG at different levels of granularity, some coarse, some finer.

5 Conclusions and Next Steps

Although they are limited to issues of time, the examples studied above suggest that an onomasiological approach gives access to textual and literary phenomena which escape tagging of textual contents alone. While the use of DAGs and threading currently requires human intervention, the output of the model, by its formality, provides a means of studying in detail the instantiation of stories as narratives, and thereby, a complement to existing approaches to literary time.

References

1. Ricœur, P.: Temps et récit. Volume 1. Éditions du seuil, Paris (1983)
2. Allen, J.F.: Towards a general theory of action and time. *Artificial Intelligence* **23** (1984) 123–154
3. Genette, G.: Figures III. Editions du Seuil, Paris (1972)
4. Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust specification of event and temporal expressions in text. In: Fifth International Workshop on Computational Semantics (IWCS-5). (2003)
5. Callaway, C., Lester, J.: Narrative prose generation. *Artificial Intelligence* **139**(2) (2002) 213–252
6. Riedl, M., Young, R.M.: Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research* **39** (2010) 217–268
7. Baldinger, K.: Sémasiologie et onomasiologie. *Revue de linguistique romane* **28** (1964) 249–272
8. Elson, D.K.: Modeling Narrative Discourse. PhD thesis, Columbia University (2012)
9. Levison, M., Lessard, G., Thomas, C., Donald, M.: The Semantic Representation of Natural Language. Bloomsbury, London (2012)
10. Mani, I.: Computational Modeling of Narrative. Morgan and Claypool, San Rafael, CA (2013)
11. Grethlein, J.: The narrative reconfiguration of time beyond Ricoeur. *Poetics Today* **31**(2) (2010) 313–329