Groundhog DAG: Representing Semantic Repetition in Literary Narratives*

Greg Lessard
French Studies
Queen's University
Canada
greg.lessard@queensu.ca

Michael Levison
School of Computing
Queen's University
Canada
levison@cs.queensu.ca

Abstract

This paper discusses the concept of *semantic repetition* in literary texts, that is, the recurrence of elements of meaning, possibly in the absence of repeated formal elements. A typology of semantic repetition is presented, as well as a framework for analysis based on the use of threaded Directed Acyclic Graphs. This model is applied to the script for the movie *Groundhog Day*. It is shown first that semantic repetition presents a number of traits not found in the case of the repetition of formal elements (letters, words, etc.). Consideration of the threaded DAG also brings to light several classes of semantic repetition, between individual nodes of a DAG, between subDAGs within a larger DAG, and between structures of subDAGs, both within and across texts. The model presented here provides a basis for the detailed study of additional literary texts at the semantic level and illustrates the tractability of the formalism used for analysis of texts of some considerable length and complexity.

1 Background

Repetition, that is, the reuse of a finite number of elements, is a fundamental characteristic of natural language. Thus, the words of a language are composed of a small number of phonemes or letters, sentences are constructed from repeated words, as well as larger collocational or syntactic chunks, and so on. Most work on repetition has been concerned with the study of such recurring formal elements, typically from the perspective of their frequency. However, it is important to recognize that a text can present not only cases in which some form recurs, as in 1(a) below, but also instances where meaning recurs, without any formal element being necessarily repeated, as in 1(b).

(1) (a) Brutus has *killed* Caesar! He has *killed* him!

(b) Brutus has *killed* Caesar! He *plunged his knife into him and our beloved leader is dead!*

It is such *semantic repetition* that concerns us here: that is, the repetition of some semantic content within a text, without there being necessarily a formal element which recurs. In particular, we wish to study semantic repetition in literary texts. This is important, since literature often brings with it the expectation that repetition is significant. To put this another way, repetition tends to be *semanticized*: its very existence 'means' something. Consider this first at the formal level. It is well-known that human language processing tends to extract meaning from sequences of forms and retain the forms themselves for only a limited time. Literary texts counteract this fading effect by several linguistic means, including physical proximity of repeated items, stress, and syntactic position. Devices such as these often carry additional information on importance or some other factor, as when an orator repeats the same word or sequence. This has been much discussed. To mention several examples among many, Jakobson (1960) refers to this as the *poetic function* of language, Genette (1972) provides a typology of narrative repetition, Tsur (2008) argues

*This research was supported in part by the Social Sciences and Humanities Research Council of Canada.

Proceedings of the Second Workshop on Computational Linguistics for Literature, pages 52–60,
Atlanta, Georgia, June 14, 2013.

that repetition is one of the devices which 'slows down' processing of text and contributes to poetic effects, Tannen (1989) gives examples of the usage of repetition in oral discourse, Okpewho (1992) shows its importance in folk literature, and Johnstone (1991) examines the role of repetition in Arabic discourse.

As we will see below, semantic repetition in literature also lends itself to semanticization. In other words, the fact that events are repeated in a narrative can be, and often is, seen not as the product of chance but rather as part of a larger pattern. The potential for this is supported by several features of meaning. First, as Stanhope et al. (1993) have shown in their work on the long-term retention of a novel, unlike formal elements, at least some semantic elements are extremely resistant to decay and can be recalled weeks and even many months later. As a result, the fading effects observed earlier for formal repetition cannot be assumed to apply in exactly the same fashion to semantic repetition: items remain accessible across entire texts and even across different texts. Second, there is in principle no upper limit on the size of semantic elements which may be repeated. At one extreme, a single character from a novel may remain in memory, along with some of the items associated with him or her. If one hears the single word *Hamlet*, what comes to mind? At the other, entire plots may be recalled. If asked to summarize the plot of *A Christmas Carol* in 100 words, most native speakers would have no difficulty in doing this. And third, by their tendency to exploit and semanticize repetition, literary texts differ from other genres, such as expository texts, whose goal is typically to present some set of information in a coherent fashion such that the same element *not* be repeated.

In light of this, our goal here is threefold: to give a sense of the diversity of semantic repetition in literary texts, including its various granularities; to propose a formal model capable of dealing with these various dimensions of semantic repetition; and to test this model against an actual text of some considerable length.

2 Events and repetition

Let us assume for the moment that semantic repetition is limited to repeated *events*, leaving aside issues of repeated qualities, entities and so on. A number of formal and semantic tools suggest themselves for dealing with this case. Within a single utterance, a neo-Davidsonian event semantics might be used, as shown in (2), where *e* represents the 'glue' which ties together the action and the agent.

$$\exists e[speak(e) \wedge agent(e) = fred(e)] \qquad (2)$$

This places the event at the centre of focus. The logical machinery behind this has been extended in various ways. For example, Hewitt (2012) proposes the use of *serial logic* to capture ordered sets of events. In addition, since events are also repeated across utterances and related to other events, as in conversations, Asher and Lascarides (2003) provides an extended logical formalism to begin to deal with this and Helbig (2006) proposes several specific functions for linking propositions, including `CAUS` (causality), `CONTR` (contrast), and `CONF` (conformity with an abstract frame). However, both approaches are applied to short spans of text and neither deals explicitly with repetition.

At a slightly higher level of granularity, Rhetorical Structure Theory (Mann and Thompson, 1988) provides a set of frameworks to describe relationships among elements of a paragraph, some of which, `Restatement` and `Elaboration` in particular, have the potential to deal with elements of repetition.[1] Work in Natural Language Generation, which has often focused on the production of longer expository texts, has also typically paid more attention to the reduction of repetition than to its production.[2] Even work on narrative generation has tended to concentrate mostly on reduction of repetition (Callaway and Lester, 2001; Callaway and Lester, 2002).

Several attempts have been made to deal with longer spans of texts, typically based on the markup of elements within a text. Most recently, Mani (2012) proposes a *Narrative Markup Language* capable of dealing with elements of repetition, but this markup is anchored to the text itself and it is unclear how such an approach could capture more abstract elements of semantic repetition. In fact, the fundamental issue

[1] For details, see `http://www.sfu.ca/rst/01intro/definitions.html`.

[2] See, however, de Rosis and Grasso (2000) who argue for the role of what they call *redundancy*.

is that semantic repetition exists across a wide range of spans, from the very smallest (both across different events and within elements of some inherently repeated activity (Tovena and Donazzan, 2008)), to the very largest, spanning multiple texts. To illustrate this, consider the following cases.

(a) A single event and the memory of the event in the mind of the perpetrator. For example, Brutus stabs Caesar, and then the next day replays the stabbing in his memory.

(b) A single event seen from the point of view of two different characters. For example, Livia sees Brutus stab Caesar, and so does Cassius.

(c) A single, perhaps complex, event, whose different facets are represented, perhaps in an interspersed fashion. Good examples of this are found in cinema, such as Eisenstein's famous bridge scene in *October*, or the Odessa steps scene in *Battleship Potemkin*, where the same images recur (such as the opening bridge, the dead horse, or the baby carriage tipping on the end of a stairway).

Examples such as these illustrate what might be called *representational repetition*, in which the same (perhaps complex) event is shown from different points of view. However, we also find examples of what might be called *class-based repetition*, in which various simple examples share a common abstract structure, as the following examples illustrate.

(d) Two sets of events in the same text represent instantiations of the same *topos*, or recurring narrative structure. For example, the Hebrew Bible contains multiple instances in which a parent favours a younger sibling over an older one. Thus, the Deity favours Abel over Cain, Abraham favours Isaac over Ishmael, Isaac favours Jacob over Esau, and so on. In these cases, we are actually faced with an abstract framework which is instantiated with different actual parameters.

(e) Two different texts represent the same abstract plot. Thus, *Pyramus and Thisbe* and *Romeo and Juliet* may both be represented by the same abstract formula, which we captures the story of star-crossed lovers whose feuding families lead to their demise.

Examples such as (d) and (e) show that at least some elements of literary repetition may only be captured by some device which permits a greater degree of abstraction than is provided by traditional devices like predicate calculus or instance-based markup. From the literary perspective, they are sometimes referred to as *topoi*, that is, recurring narrative sequences.[3] However, as formulated in most literary analyses, the notion of topos has several shortcomings. First, definitions tend to be informal.[4] Second, the granularity of topoi is unclear. One might express a given topos in very general terms or quite specifically.

Our goal here is to build on the insights of literary theory regarding the *meaning* of literary texts, while retaining some level of formalism. To do this, we need first to respect the empirical richness of literary texts. As the examples above show, simple two-line examples are not sufficient to show the true complexity of semantic repetition. Accordingly, we have chosen as our corpus an entire movie script, described below. Second, in the case of semantic repetition, we need a formalism capable of captureing various levels of granularity, from quite fine to very general, and which shows not just differences of point of view, but elements of class inclusion. To do accomplish this, we have adopted the formalism described in Levison et al. (2012), based on a functional representation of meaning elements by means of *semantic expressions*.[5] When combined with the use of threaded Directed Acyclic Graphs, discussed below, this formalism permits the representation of elements of meaning at various levels of granularity,

3 *Groundhog Day*

To illustrate the phenomenon of semantic repetition, we have created a formal analysis of the screenplay for the popular movie *Groundhog Day* (henceforth,

[3] A detailed list of topoi, together with examples, may be found in `http://satorbase.org`.

[4] See Lessard et al. (2004) for one attempt at formalization. Note also that the concept of topos shares features with the concept of *scripts* (Schank and Abelson, 1977), which has been formalized to some degree.

[5] The formalism is inspired by the Haskell programming language (Bird, 1998).

GH).[6] Because of its plot structure, discussed below, the script represents arguably an extreme case of semantic repetition and thus a good test of the proposed model of semantic repetition.

GH recounts the story of Phil Connors, an egocentric weatherman, who has been sent with his producer, Rita, and cameraman Larry, to cover the annual February 2 event at Punxsutawney, Pennsylvania, where a groundhog (Punxsutawney Phil), by seeing or not seeing his shadow, provides a prediction on the number of weeks remaining in winter. Displeased at such a lowly assignment, Connors behaves badly to all. However, on waking up the next day, he discovers that it is still February 2, and the day unfolds as it had previously. In the many subsequent iterations of the day, Connors discovers the possibilities inherent in there being no consequences to his acts, the advantages of being able to perfect the elements of a seduction by repeated trials, and finally, the value of altruism and love. At this point, after many iterations, the cycle is broken, and Phil and Rita, now in love, greet February 3.[7]

4 Directed Acyclic Graphs

To capture the various elements of granularity in the GH script, we make use of the well-known distinction in literary theory between two perspectives on a narrative. The *fabula* or *histoire* is the information on which the narrative is based; the *sjuzhet* or *récit* is a particular telling (Bal, 1985; Genette, 1983). In our model, we represent the former, which we shall term a *story*, by a *Directed Acyclic Graph*, henceforth DAG. A directed graph is a collection of nodes linked by unidirectional paths. In an acyclic graph, no sequence of paths may link back to a node already visited. In technical terms, the dependency relation portrayed by the graph is transitive, irreflexive and antisymmetric. Within the DAG, nodes denote pieces of the meaning, perhaps at different levels of granularity, and directed paths which indicate the dependence of one node upon another. By dependence, we mean that subsequent nodes in the DAG make use of information present on previous nodes. In a finer analysis, the nature of the various dependencies might be sub-divided into subclasses like logical dependency, temporal dependency, and so on, but we will not do that here.

As noted earlier, we represent the meanings carried by the nodes of a DAG by means of *semantic expressions*. So, for example, given the semantic entities `phil` and `rita`, and the action `meet`, the expression `meet(phil, rita)` represents a meeting between the two characters in the film. This expression represents what is called, in the framework used, a *completion*. Although the functional representation used permits the representation of semantic niceties like temporal relations and definiteness, the model used here does not include them. In the analysis here, each semantic expression corresponds to one node of the DAG. Of course, such a model may vary in granularity. At one extreme, the entire script could be represented by a single expression (as in `improve(phil)`. At the other, each small event could form the basis of a semantic expression. For the purposes of the present analysis, we have adopted an intermediate granularity.[8]

Each element of the functional representation is drawn from a *semantic lexicon* composed of a formal specification and an informal one, which provides a basic-level textual output, as shown by the following examples:

```
meet :: (entity, entity)
      -> completion
meet(x,y) =
      "[x] meets [y]"
```

where the first line shows the formal specification and the second line the informal one. The sequence of semantic expressions, when used to call the informal representations, thus provides the gist of the script, or alternatively, can be used to drive a natural language generation environment. In addition, since the elements of the DAG are formally specified in the semantic lexicon, they may be analyzed or further manipulated by graph manipulation software. To take a trivial case, the transitive closure of a DAG might be calculated.

[6] It should be noted that this screenplay, which may be found online at `http://www.dailyscript.com/scripts/groundhogday.pdf`, diverges in some respects from the film. It contains some scenes which do not appear in the film, and it does not contain some others which do appear in the film.

[7] A fuller synopsis can be found at `http://www.imdb.com/title/tt0107048/synopsis`.

[8] A fuller discussion of these issues may be found in Levison and Lessard (2012).

5 Threads and threading of a DAG

A particular telling of a story, which we call here the *narrative*, may be conceived of as a particular traversal of the DAG. To designate this, we make use of the concept of *threading*. Threads are simply sequences of nodes and we often display them in the diagram of a DAG by a dotted line through the nodes. A thread need not follow the edges of the DAG, nor need it be acyclic. In other words, the same thread may traverse the same node more than once. The ordering of the threads of a narrative is assumed to correspond to narrative time. The various segments in our diagrams are numbered. Threads may traverse some but not necessarily all nodes of the DAG.

It should be noted that a particular DAG may give rise to numerous possible threadings. So, for example, a story may be told in chronological order ("Once upon a time, there was a beautiful princess ... she was kidnapped by an evil wizard ... a handsome prince rescued her ... they lived happily ever after."), or in reverse ("The prince and the princess were preparing for their wedding ... this was the outcome of his rescue of her ... she had been kidnapped..."). Furthermore, a DAG may be threaded to capture not just some telling of the narrative, but also in terms of the point of view of some character, the states of some object in the narrative, or the representation of space or description of places or characters.

We will apply this conceptual machinery to the analysis semantic repetition in the GH script.

6 Analysis

At an abstract level, the relationships behind GH (that is, the *story*) may be represented by three nodes joined by solid edges, which show the semantic dependencies among the nodes, as shown in Figure 1. The first sets the scene by placing Phil in Punxsutawney, the second represents Phil's recursive actions during his endless series of February 2's, and the third represents his escape from recursion.

At the opposite extreme of granularity, it is possible to show the GH DAG with a thread traversing fine-grained nodes, each represented by a semantic expression. This representation, which contains 172 nodes and 171 edges, is far too large to fit onto a page. It may be viewed in its entirety at `http://tinyurl.com/awsb4x6`. As noted

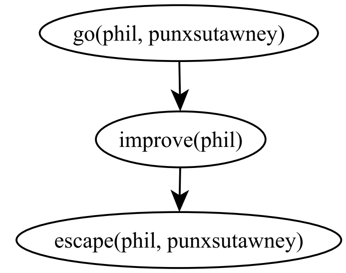

Figure 1: The most abstract DAG for GH

above, the segments of the thread are numbered and dotted. Following them in order thus recounts the semantic representation of the GH narrative at a relatively fine level of granularity. Between these two extremes of the abstract DAG and the linear threading, we will now examine several issues of semantic repetition.

6.1 Repetition as return of threads to a node

The simplest form of semantic repetition takes the form of a thread passing through some node more than once. Figure 2 provides a simple case of this.

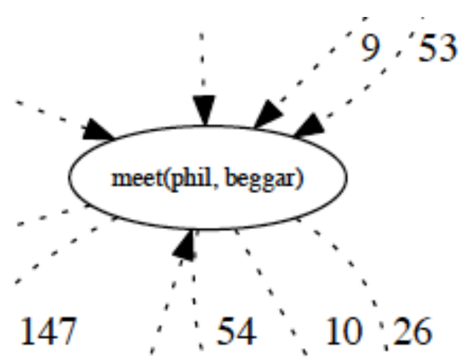

Figure 2: A thread passing multiple times through the same node

Thus, Phil meets a beggar at several points in the narrative (threads 9, 53, 146), with various outcomes, including ignoring the beggar (threads 10, 26, 54) and helping him (thread 147). Despite this help, the beggar dies (thread 148), but Phil is given the opportunity to replay this sequence (thread 149), choosing then to feed the beggar (thread 150).

6.2 DAGs and subDAGs

Consideration of the entire GH threading shows not just return of the thread to a single node, but also constellations of nodes which 'hang together'. In some cases, this is based on common membership of the nodes in some class of events. One example of this is provided by Phil's various attempts at suicide. Since Phil returns to life after each suicide, each suicide attempt (a toaster dropped into a bathtub, leaping from a tall building, walking in front of a bus, and so on) shares with the others only membership in the class of suicide events. This state of affairs may be captured by including each of these nodes within a local subDAG, which itself represents a subnode of the larger DAG. So, for example, we could represent the local subDAG here by means of the semantic expression `attempt(phil, suicide)`. Such subDAGs may be further refined or combined, similar to the concept of stepwise refinement found in computer programming.

In the case of the various suicide attempts, it is important to note that the various attempts show no dependency among themselves, and no order among them is required, beyond that imposed by a particular threading. This may be represented as follows:

```
kill(phil, phil, with(electricity))
kill(phil, phil, with(jump))
```

and so on. A similar example is found in Phil's attempts to improve himself, which involve learning Italian, music, sculpture and medicine, among other things.

However, we also find instances in which several nodes within a subDAG do show dependency relations within a common subDAG. So, for example, when Phil meets Rita at a bar, the same sequence is followed: he buys her a drink, they make a toast, and they discuss Rita's previous education, as can be seen in Figure 3.

Note that both temporal and logical dependence exists between two of the nodes (Phil must buy the drink in order for them to make a toast). There is no dependence between these two and the discussion of Rita's education, but the threading may indicate a temporal order.

Mixed models are also possible, in which some elements of a subDAG show dependency while others do not, as in the case where Phil awakens to the fact that his acts have no long-term consequences. In one reaction to this, he robs a bank, buys a car and tours the town. Each of these steps depends on the previous one. However, he also gets a tattoo and throws a party, both of which are independent of each other and of the others. However, together, all these elements constitute the subDAG of exploring the absence of consequences.

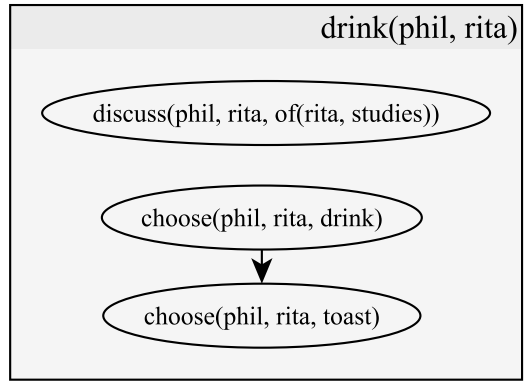

Figure 3: The subDAG for Phil and Rita at the bar

6.3 Parametrized subDAGs

In the presentation so far, we have treated the semantic expressions within nodes as constants. However, examination of the GH DAG brings to light several instances in which some part of the DAG is repeated with one or more elements replaced systematically with different ones. One illustration of this may be found in Phil's various reportings of the events at Gobbler's Knob, when the groundhog appears. Over the course of the narrative, he is first glib and sarcastic, then confused, then professional, then learned, poetic, and finally profound. This might be represented by five distinct copies of the part.

```
describe(phil, groundhog, ironic)
describe(phil, groundhog, confused)
```

and so on. However, given the similarity between the five nodes, it would be more efficient to create a single, separated, copy containing parameters, which could be instantiated in each of the five places with the parameters replaced by the appropriate variants.

A similar series of parameters may be found elsewhere in GH, for example, when Phil greets the man on the stairs of the B&B first ironically, then angrily, and finally with good humour, in Italian. Or again, at a more complex level, we find a series of instances where Phil learns some new skill (French poetry, piano, Italian, sculpture,...) and subsequently applies it. This is illustrated by two typical subDAGs in Figure 4.

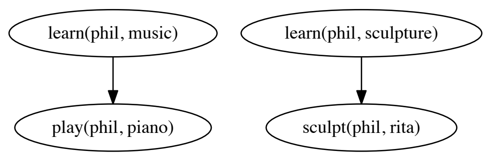

Figure 4: Learning and implementation

Each of these series forms a sequence such as:

```
improve(phil, altruism, 0)
improve(phil, altruism, 1)
```

and so on, where the third parameter indicates Phil's progression along the scale of character development. This particular series provides a means of capturing each particular state in Phil's evolution from egotist to altruist.

Note however that Phil's moral development does not progress through different areas of his life, one series at a time. In other words, he does not first change from a sarcastic to a poetic reporter, then grow from an egotist to an altruist in the community, then make the transformation from a seducer to an attentive lover, and so on. Rather, his personal improvement happens more or less at the same pace across different facets of his life, reflecting his overall personal growth, although evidence of this might be drawn first from one and then from another of his activities.

6.4 Parallel DAGs

In the discussion to this point, we have been concerned with repetition within a single subDAG. However, in GH, we also find instances where one subDAG shows an architectural similarity to another. This similarity can be construed as a sort of high-level repetition. For example, while on location in Punxsutawney, Phil meets and seduces Nancy, a woman from Punxsutawney. At the same time, he attempts to seduce Rita, his producer.

In both cases, Phil makes an initial attempt and is rebuffed, by both Nancy and Rita. Undaunted, he then seeks more information about both, determining Nancy's name and obtaining enough information to pass as a former high school classmate, and determining that Rita drinks tequila with lime, that she toasts world peace, and that she likes French poetry. He then uses the information about Nancy to seduce her, but the same tactic is unsuccessful with Rita.

The two parallel subDAGs may be represented by a higher-level subDAG where almost all the individual elements change from case to case, with only the general framework remaining. This might be expressed schematically as follows:

```
experiment(x,y) =
slist(
  meet(x,y)
  learn(x, of(y, characteristics)))
```

and so on.

Applied within a single narrative, such an approach deals with the sort of parallel cases seen here. Applied across narratives, it gives rise to texts seen as 'telling the same story', like *Romeo and Juliet* mentioned earlier. At an even more abstract level, it provides a means of modelling parodies, or works based on some previous model. Think of Joyce's *Ullysses*, in which Stephen Daedalus' peregrinations around Dublin represent a parellel to Homer's *Odyssey*.

6.5 Connections between subDAGs

We now have a means of representing semantic repetition at both the low level, of individual nodes of a DAG, as well as within and across DAGs. However, we have left unspecified an important point, to which we now return. Earlier, we showed that individual nodes may contain subDAGs of interior nodes, up to some indefinite level of complexity. This varying granularity provides a model for different degrees of detail in the recounting of a story, between the highest-level and coarsest summary, to the finest detail. Consider now the following case from GH. Each day, Phil wakes up, hears the same song on the radio, followed by inane banter from two disc jockeys. At the level of the DAG, this may be represented as an overarching node which contains two interior nodes, as shown formulaically here:

```
wakeup(phil) = slist(
  hear(phil, song)
  hear(phil, dj_banter))
```

and graphically in Figure 5.

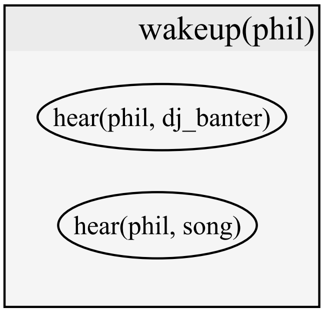

Figure 5: Part of the DAG for Phil's waking up

However, the actual threading of this higher-level nodes and its interior nodes in the narrative varies over the course of the narration, as shown in Figure 6.

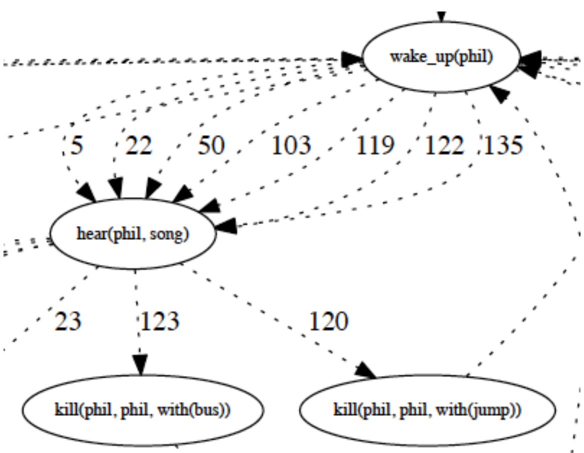

Figure 6: The threading of Phil's waking up

Thus, in threads 5, 22, 50, 103, 119, 122 and 135, Phil's waking up is followed by his hearing of the song, but in thread 36, Phil's waking up is followed immediately by the DJ banter. Similarly, threads 6, 23, 51 and 104 join the hearing of the song with the hearing of the banter, but in the case of threads 120 and 123, the recounting of Phil's hearing of the song is followed directly by suicide attempts, with no mention of the banter. In both these cases, we can presume that the DAG remains constant, but the threading represents either a complete traversal of all the interior nodes, or, typically later in the narrative, narrative 'shortcuts' which indicate the entire `wakeup` DAG by explicitly mentioning only some elements. Such shortcuts may be found in most narratives. For example, subsequent references to a known character or event may be reduced to the minimum, since a simple mention reactivates the entire reference. Conversely, the exploration of interior nodes rather than higher-level ones (in other words, providing more detail) may produce an effect of *slowdown* (Bal, 1985).

In the case of semantic repetition, shortcuts like those just described demonstrate that not only can repetition occur in the absence of repeated formal elements, but even in the absence of explicitly repeated semantic elements. At the extreme, the activation of a higher-level node by reference to an interior node provides a model for literary allusions, perhaps the most subtle type of repetition, where some element in one text activates a reference to another.

7 Conclusions and next steps

The series of examples presented here provide evidence for the existence of semantic repetition at both the atomic and structural levels. They also show that these can be captured by a model which permits various levels of granularity, from atomic semantic expressions to higher-level subDAGs and DAGs. It must be admitted however that, at this stage of the research, only human intelligence has permitted the identification of semantic repetition in its various forms. In an ideal world, a computer program might be capable of arriving at the same judgments. Work such as Chambers and Jurafsky (2008) or Hobbs et al. (1993) might provide a good starting point for this. In the meantime, we believe that there is value in continuing the *meaning-first* perspective illustrated here, as a complement to the more usual text-first analyses. When combined with a user-friendly formalism, this approach would go some way to bridging the divide between computer scientists and literary specialists in their analysis of literary texts.

References

Nicholas Asher and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press, Cambridge.

Mieke Bal. 1985. *Narratology: introduction to the theory of narrative*. University of Toronto Press, Toronto.

Richard Bird. 1998. *Introduction to functional programming using Haskell*. Prentice-Hall, London, 2nd edition.

Charles Callaway and James Lester. 2001. Evaluating the effects of natural language generation techniques on reader satisfaction. In *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society*, pages 164–169.

Charles Callaway and James Lester. 2002. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797.

Fiorella de Rosis and Floriana Grasso. 2000. Affective natural language generation. In A.M. Paiva, editor, *Affective instructions*, pages 204–218. Springer, Berlin.

Gérard Genette. 1972. *Figures III*. Editions du Seuil, Paris.

Gérard Genette. 1983. *Nouveau discours du récit*. Editions du Seuil, Paris.

Hermann Helbig. 2006. *Knowledge representation and the semantics of natural language*. Springer, Berlin.

Simon Hewitt. 2012. The logic of finite order. *Notre Dame Journal of Formal Logic*, 53(3):297–318.

Jerry R. Hobbs, Mark E. Stickel, Douglas E. Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63:69–142.

Roman Jakobson. 1960. Linguistics and poetics. In Thomas A Sebeok, editor, *Style in language*, pages 350–377. MIT, Cambridge, Mass.

Barbara Johnstone. 1991. *Repetition in Arabic discourse: paradigms, syntagms, and the ecology of language*. J. Benjamins, Amsterdam.

Greg Lessard, Stéfan Sinclair, Max Vernet, François Rouget, Elisabeth Zawisza, Louis-Émile Fromet de Rosnay, and Élise Blumet. 2004. Pour une recherche semi-automatisée des topoï narratifs. In P. Enjalbert and M. Gaio, editors, *Approches sémantiques du document électronique*, pages 113–130. Europia, Paris.

Michael Levison and Greg Lessard. 2012. Is this a DAG that I see before me? An onomasiological approach to narrative analysis and generation. In Mark Finlayson, editor, *The Third Workshop on Computational Models of Narrative*, pages 134–141, LREC Conference, Istanbul.

Michael Levison, Greg Lessard, Craig Thomas, and Matthew Donald. 2012. *The Semantic Representation of Natural Language*. Bloomsbury, London.

Inderjeet Mani. 2012. *Computational Modelling of Narrative*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

William C. Mann and Sandra Thompson. 1988. Rhetorical Structure Theory: toward a functional theory of text organization. *Text*, 8(3):243–281.

Isidore Okpewho. 1992. *African oral literature: backgrounds, character, and continuity*. Indiana University Press, Bloomington.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Lawerence Erlbaum Associates, Hillsdale, NJ.

Nicola Stanhope, Gillian Cohen, and Martin Conway. 1993. Very long-term retention of a novel. *Applied Cognitive Psychology*, 7:239–256.

Deborah Tannen. 1989. *Talking voices: repetition, dialogue, and imagery in conversational discourse*. Cambridge University Press, Cambridge.

Lucia M. Tovena and Marta Donazzan. 2008. On ways of repeating. *Recherches linguistiques de Vincennes*, 37:85–112.

Reuven Tsur. 2008. *Toward a theory of cognitive poetics*. Sussex Academic Press, Brighton, 2nd edition.