6th Workshop on Computational Models of Narrative

CMN 2015, May 26–28, 2015, Atlanta, U.S.A.

Edited by

Mark A. Finlayson
Ben Miller
Antonio Lieto
Remi Ronfard

OASICS

Editors

Mark A. Finlayson
Florida International University
School of Computing and Information Sciences
Miami, FL, U.S.A.
`markaf@fiu.edu`

Ben Miller
Georgia State University
Department of English
Atlanta, Georgia, U.S.A.
`miller@gsu.edu`

Antonio Lieto
Department of Computer Science
University of Torino
Torino, Italy
`lieto@di.unito.it`

Rémi Ronfard
Centre de recherche INRIA
IMAGINE Laboratoire Jean Kuntzmann
Grenoble, France
`remi.ronfard@inria.fr`

ACM Classification 1998
F.4.1 Knowledge Representation Formalisms and Methods, I.2: Artificial Intelligence, I.2.0 General/Cognitive simulation, I.2.7 Natural Language Processing, I.6 Simulation and Modeling, I.2.8 Problem Solving, Control Methods, and Search, I.2.11 Distributed Artificial Intelligence, J.4 Psychology, J.5 Literature

ISBN 978-3-939897-93-4

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at http://www.dagstuhl.de/dagpub/978-3-939897-93-4.

Publication date
July, 2015

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Digital Object Identifier: 10.4230/OASIcs.CMN.2015.i

ISBN 978-3-939897-93-4 **ISSN 2190-6807** **http://www.dagstuhl.de/oasics**

OASIcs – OpenAccess Series in Informatics

OASIcs aims at a suitable publication venue to publish peer-reviewed collections of papers emerging from a scientific event. OASIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

ISSN 2190-6807

www.dagstuhl.de/oasics

Contents

Invited Talk

Regular Papers

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard
OASICS OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

Welcome to the Sixth Workshop on Computational Models of Narrative. This year finds us co-located with the Third Annual Conference of Advanced in Cognitive Systems (CogSys 2015). This association made it appropriate to have a special focus on the intersection of cognitive systems and narrative. This intersection is rich and broad, covering the gamut from psychological and cognitive impact of narratives to our ability to model narrative responses computationally. Papers contributed to this volume tackle questions of narrative analysis in the domains of medical information and journalism, and of various story generation systems and frameworks. They look to extend prior paradigms, in one case connecting event segmentation theory to the computational modeling of narrative, and in another, proposing a model for synthesizing temporal, ontological, and psychological aspects of story. And they report on experiments such as the application of syntactic and semantic feature detection to the exploration of higher-level storytelling tropes such as romantic love and animacy.

Interest in and submissions to the CMN workshop remain robust. This year we received 22 submissions; of these 6 were declined. In keeping with our goal of inclusiveness, 16 papers were accepted, some on condition of revision. None of these revised papers were declined after revision, although one paper was withdrawn. Including one additional keynote abstract brings the total number of published works in this proceedings to 16. Over seven years, six meetings, and five volumes of proceedings, the CMN workshop series has published 118 works. This sustained pace demonstrates the consistent relevance of the workshop series and its contributions to the field.

Last year, in an effort to ensure the longevity and continued vitality of the workshop series, a transition period began from Mark Finlayson being the primary steward to a more formal organizational structure. A steering committee is being established comprised of former organizers and co-organizers of the workshop. We began a 'staged' organization arrangement, where those who volunteer to be lead organizer of the workshop in year X are co-organizing the workshop in year X-1. This arrangement led to this year's workshop being organized by the committee of Mark Finlayson, Ben Miller, Remi Ronfard, and Antonio Lieto. This structure has helped the new organizers learn the ropes and lent continuity to the series.

We are also please to announce the winner of our best paper award. The award and a $250 check goes to Mr. Folgert Karsdrop for his paper "Animacy Detection in Stories", co-authored with Marten van der Meulen, Theo Meder, and Antal van den Bosch.

Many thanks to our generous sponsors without whom this year's workshop would not have been possible: The Georgia Institute of Technology has graciously provided the workshop venue, and supplemental funding was provided by the Department of English and the Creative Media Industries Institute at Georgia State University.

Mark Finlayson & Ben Miller

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard
OASICS OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

List of Authors

Adam A. Amos-Binks
North Carolina State University
Raleigh, NC, U.S.A.
aaamosbi@ncsu.edu

Tory S. Anderson
Georgia Institute of Technology
Atlanta, GA, U.S.A.
torys.anderson@gatech.edu

Camille Barot
North Carolina State University
Raleigh, NC, U.S.A.
cbarot@ncsu.edu

Steve Battle
Sysemia Ltd
Bristol, United Kingdom
steve.battle@sysemia.co.uk

Fritz Breithaupt
Indiana University
Bloomington, IN, U.S.A.
fbreitha@indiana.edu

Eleanor Brower
Indiana University
Bloomington, IN, U.S.A.

Beth Cardier
Sirius Beta Inc.
Virginia Beach, VA, U.S.A.
bethcardier@sirius-beta.com

David A. Caswell Structured Stories
Los Angeles, CA, U.S.A.
david@structuredstories.com

Markus Eger
North Carolina State University
Raleigh, NC, U.S.A.
meger@ncsu.edu

George Fantry
University of Maryland,
Baltimore, MD, U.S.A.
Gfantry@medicine.umaryland.edu

Pablo Gervás
Universidad Complutense de Madrid
Madrid, Spain
pgervas@ucm.es

Shakthidhar Gopavaram
Georgia State University
Atlanta, GA, U.S.A
sgopavaram1@cs.gsu.edu

Folgert Karsdorp
Meertens Institute
Amsterdam, The Netherlands
folgert.karsdorp@meertens.knaw.nl

Mike Kestemont
University of Antwerp
Antwerp, Belgium
mike.kestemont@uantwerp.be

Arnav Jhala
University of California
Santa Cruz, CA, U.S.A.
jhala@soe.ucsc.edu

Bruce Jarrell
University of Maryland
Baltimore, MD, U.S.A.
bjarrell@umaryland.edu

Sarah Harmon University of California
Santa Cruz, CA, U.S.A.
smharmon@ucsc.edu

Carlos León
Universidad Complutense de Madrid
Madrid, Spain
cleon@ucm.es

Marjorie McShane
Rensselaer Polytechnic Institute
Troy, NY, U.S.A.
mcsham2@rpi.edu

Gonzalo Méndez
Universidad Complutense de Madrid
Madrid, Spain
gmendez@ucm.es

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard
OASICS OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Theo Meder
Meertens Institute
Amsterdam, The Netherlands
theo.meder@meertens.knaw.nl

Ben Miller
Georgia State University
Atlanta, GA, U.S.A
miller@gsu.edu

Sergei Nirenburg
Rensselaer Polytechnic Institute
Troy, NY, U.S.A.
nirens@rpi.edu

Jennifer Olive
Georgia State University
Atlanta, GA, U.S.A
jolive1@gsu.edu

Julian Padget
University of Bath
Bath, United Kingdom
mrt32@bath.ac.uk

Christof Schöch
University of Würzburg
Würzburg, Germany
c.schoech@gmail.com

Ayush Shrestha
Georgia State University
Atlanta, GA, U.S.A
ayush.shrestha@gmail.com

Nicolas Szilas
University of Geneva
Genève 4 Switzerland
Nicolas.Szilas@unige.ch

Matt Thompson
University of Bath
Bath, United Kingdom
asjap@bath.ac.uk

Marten van der Meulen
Meertens Institute
Amsterdam, The Netherlands
marten.van.der.meulen@meertens.knaw.nl

Antal van den Bosch
Radboud University
Nijmegen, The Netherlands
a.vandenbosch@let.ru.nl

R. Michael Young
North Carolina State University
Raleigh, NC, U.S.A.
rmyoung@ncsu.edu

Sarah Whaley
Indiana University
Bloomington, IN, U.S.A.

David R. Winer
North Carolina State University
Raleigh, NC, U.S.A.
drwiner@ncsu.edu

Patrick H. Winston
Massachusetts Institute of Technology
Cambridge, MA, U.S.A.
phw@mit.edu

Tell Me a Story: Toward More Expressive and Coherent Computational Narratives

Janet H. Murray

Georgia Tech
Atlanta, GA, U.S.A
jmurray@gatech.edu

Abstract

Since narrative is a foundational framework for the on-going co-evolution of human cognition and culture, the advent of computation as a new medium for representing narratives offers the promise of ratcheting up human understanding and expressive power, just as previous media of representation like language and writing have done. But digital representation often produces artifacts that are story-like but not really stories, leaving open the question of how we can make use of computational models of narrative to expand our capacity for shared meaning-making. I will address this problem by looking at the complementary strengths and weaknesses of simulation making, game design, and storytelling as cultural abstraction systems, and suggest some directions for incorporating richer story structures into research on computational narratives.

1998 ACM Subject Classification I.2.0 General

Keywords and phrases computational models of narrative

Digital Object Identifier 10.4230/OASIcs.CMN.2015.1

Category Invited Talk

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 1–1
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

From Episodic Memory to Narrative in a Cognitive Architecture

Tory S. Anderson

Georgia Institute of Technology
North Avenue, Atlanta, GA 30332, USA
torys.anderson@gatech.edu

Abstract

Human experiences are stored in episodic memory and are the basis for developing semantic narrative structures and many of the narratives we continually compose. Episodic memory has only recently been recognized as a necessary module in general cognitive architectures and little work has been done to examine how the data stored by these modules may be formulated as narrative structures. This paper regards episodic memory as fundamental to narrative intelligence and considers the gap between simple episodic memory representations and narrative structures, and proposes an approach to generating basic narratives from episodic sequences. An approach is outlined considering the Soar general cognitive architecture and Zacks' Event Segmentation Theory.

1998 ACM Subject Classification I.2.0 Cognitive Simulation, J.4 Psychology, J.5 Literature

Keywords and phrases Narrative, Episodic Memory, Cognitive Architecture, Event Segmentation

Digital Object Identifier 10.4230/OASIcs.CMN.2015.2

1 Introduction

Since Tulving's pioneering work on episodic memory [33] it has become apparent that any general model of human cognition must account for memory for temporally and causally situated data just as well as memory for the general facts of semantic memory. It has been observed that we perform extensive narrative sense-making over the data we experience in an effort to gather meaning from our raw experiences [9]; this activity is central to our lives. This ability to cast our experience in narrative terms has been referred to as narrative intelligence [20, 3] and develops through our formative years. Sharing features of both narrative comprehension and narrative generation, narrative intelligence is important to our planning, social interaction, and coping with challenges [23]. This has led to a surge of interest in narrative processes for artificial intelligence [20]; nonetheless, cognitive architectures aimed at modeling human intelligence have been slow to implement support for episodic memory and have as-yet showed few signs of approaching narrative cognition.

1.1 Narrative Intelligence, Comprehension, and Generation

Mateas' definition of narrative intelligence has already been invoked as a guiding concept: the ability to cast our experience in narrative terms. We are here concerned with this sophisticated process, which simultaneously draws from and defies frameworks that attempt to delineate story comprehension from story generation. The input to our model is a stream of experiential data; the process of parsing and selecting from this data, for which Event Segmentation

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 2–11
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

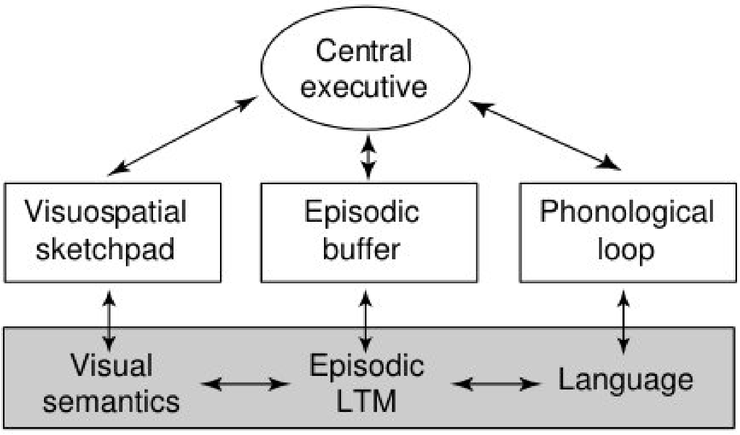


Figure 1 Baddeley's revised working memory model, including the episodic buffer [2].

Theory (EST) will be applied, can be seen as narrative comprehension insomuch as top-down processing occurs to recognize matching narrative patterns. Inasmuch as bottom-up processing is performed upon the received data, a process central to the gating mechanisms of EST, it is similar to some plan-based narrative generation systems which receive a repertoire of actions and use that repertoire to generate a sequence of states as a narrative (e.g. [29]). This reciprocation between narrative comprehension and narrative generation bears striking similarity to the driving tension of cognitive narrative pointed out by Ochs and Capps in their landmark study of personal narratives, described as "the oscillation between narrators' yearning for coherence of life experience and their yearning for authenticity" [23, p. 24]. For cognitive narrative the distinction between narrative comprehension and narrative generation, principle to some notions of intelligence for narrative [17], may need reevaluation.

Importantly, while the joint pair of narrative comprehension and generation are of major relevance to this paper, the distinct process of *story telling*, by which narratives are prepared and committed via some media for purposes that include communication, falls beyond our consideration of cognitive narrative and can be regarded as an activity occurring subsequent to (and using the products of) the processes here proposed.

2 Memory, Segmentation, and Narrative

Narrative exists in the human mind as a particularly important form of mental technology. It's utilization includes experiential sense-making, imputing of causality, categorization and evaluation of events, complex communication, and planning [10]. Narrative cognition is inextricably involved with human memory, particularly the episodic and semantic long-term memory systems. Semantic memory supplies the scripts, schemas, and genres by which top-down processes influence narrative cognition [32, 27], and so plays a vital role in mature narrative intelligence. Evidence from developing narrative intelligence within children suggests that the acquisition of these semantic structures is one of the significant forms of progress as children grow [34][23, ch. 2]. However, the same evidence indicates that however poor, some degree of narrative ability precedes the significant acquisition of semantic narrative structures and that one of the functions of increasing experience is the construction of the scripts and schema that will allow for improved top-down contributions to narrative intelligence. This suggests that narrative intelligence may begin with episodic memory before being augmented with contributions from semantic memory.

Episodic memory is the system responsible for storage of both personal experiences and any other time-situated events attended to second-hand, for example through media or personally communicated stories. It is also implicated for prospective memory used to consider the future [31]. As a distinct memory system it was first proposed by Endel Tulving in 1972 [33]; since that time it has been widely researched. Of particular note is work by Baddeley, who augmented his 1970 model of working memory with an episodic buffer (Figure 1). This episodic buffer was proposed for use in short-term memory complementary to the conventionally understood episodic long-term memory [2]. The role of Baddeley's short-term episodic buffer is as a holding area for retrieved episodes to be integrated cross-modally with data from other sources, such as perception or semantic processing. From a narrative perspective, this may be where stories are constructed through blending with other elements in working and semantic memory, and is likely where narratives are manipulated for many of the afore-mentioned functions of narrative cognition.

The term "episode" excites a notion of scene, events, and change that would seem naturally compatible with most definitions of narrative. However, event recognition itself is an ongoing challenge in computer science. In practice, implementations of episodic memory usually operate as the storage and chronological indexing of system states. In essence, these systems take a snapshot of each state and give it a time label. While narratively intelligent humans are capable of looking at a photo (e.g. of a sport scene) and reconstructing a narrative situation to describe the events surrounding the scene, for these computational systems there has been no obvious way to produce from a life-long sequence of such snapshots a discrete set of narratives.

2.1 Event Segmentation Theory

Event Segmentation Theory (EST) [35, 13, 27] suggests an approach to the problem of dividing a non-delineated sequence of states into events that could become the constituents of narratives. In humans, event segmentation is an ongoing process occurring simultaneously at multiple time/action granularities. According to EST, event segmentation occurs as an effect of ongoing perceptual prediction. During the process of perception two structures participate in parsing the situation and forming predictions: long-term knowledge is brought to bear in the form of event schemata, which are similar to Schanks' and Abelson's scripts [32] and represent the way actions or events normally unfold in similar situations; and working-memory is brought to bear by event models, which are an interpretation of the specific situation at hand. In addition, behavioral models may be used so that predictions can be made based on the presumed goals of the actors in a situation, and world models that account for physical expectations (e.g. the trajectory of an object in free motion). The interplay between the semantic and episodic long-term memory systems in this process is cyclical: semantic memory provides the structures and models to help make episodes from experience, while these episodes are committed to episodic memory where, over time, they help distill further knowledge of semantic structures.

As perception occurs, the mind selects from its knowledge of usual event schemas and uses assumptions about the goals and processes at work in the attended situation to generate expectations of what will happen next. As long as these predictions are mostly fulfilled, the current event model is assumed to continue and no segmentation occurs. However, when the predictions are wrong by some margin of significance, the current event is considered to end and a new event begin in the process of selecting or generating a new event model. These explanations of event segmentation have been supported by evidence from studies of segmentation of event boundaries in written and video narratives [35]. Narratives are

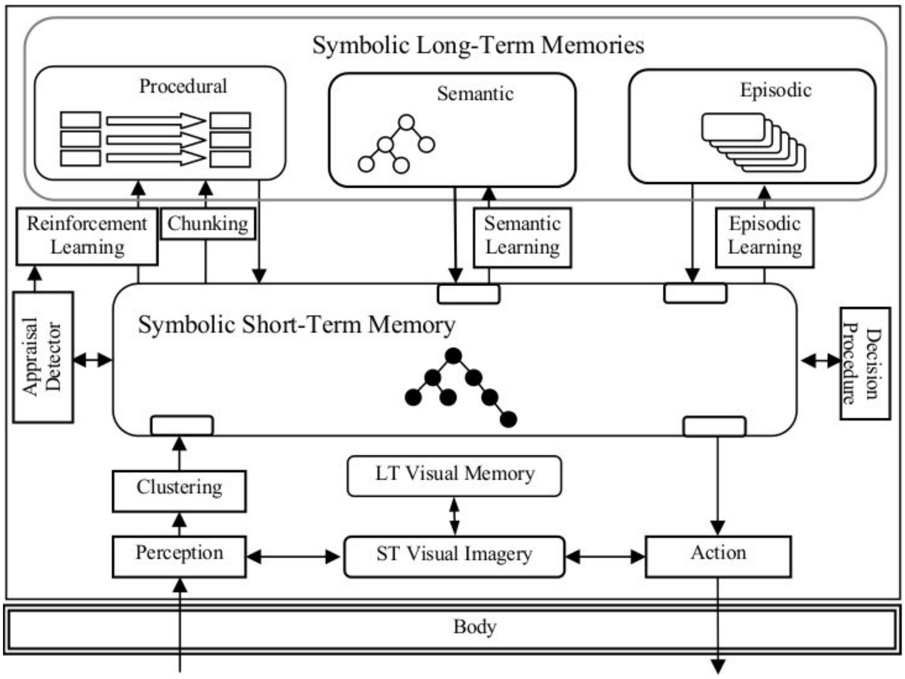


Figure 2 The Soar cognitive architecture [14].

constructed as segmentation occurs at broader granularities over episodic memory, to the point of eventually contributing to production of the life-long autobiographical memories that "make up our own personal narrative of who we are and what we have experienced" [27, ch. 8].

3 An Approach with the Soar Cognitive Architecture

Although it has been explored in a neural network framework [28], EST has yet to be applied in a symbolic architecture. Soar [15] (see Figure 2) is a general cognitive architecture with development overseen by John Laird and is one of the most popular cognitive architectures in current use, with deployments ranging from robotic intelligence to complex battlefield simulation to military training of human soldiers. In addition to an AI system, Soar represents a theory of general human cognition [22]. Soar is a rule-based system in which perception is represented as a graph structure in either working memory or long-term memory. Soar is also agent-based, meaning that instances of Soar run as individual agents independent of, but often interacting with, each other. A given application can call upon large numbers of Soar agents, each running as its own process with its own long-term memory and working memory systems. Soar agents make decisions based on the matching of rules, which depend on the agent's perception of the current state of the world and of its personal state. As a symbolic architecture Soar is well-suited to capturing top-down information such as explicit scripts or subjects of high-level complexity like narrative, whereas it can be difficult to obtain narrative training sets that are both suitably representative and sufficiently sizable for the needs of connectionist models.

Soar's episodic memory modules (epmem) depicted in the top right corner of Figure 2 were added relatively recently and are our central focus. Soar's epmem works by storing snapshots of the working memory state (i.e. the Soar agent's awareness) at each time step, attaching to each snapshot a unique index representing the time of the memory. Once Soar has recalled an episodic memory it is possible to increment forward or backward through the neighboring episodes. Retrieval of episodic memory occurs as queries are issued searching for matching or partially matching features in the graph-structure knowledge representation. Results are given a match score based on how much of the query-graph matches the graphs in an episode, and the best match is returned.

The aim of this project is to outline the addition of rudimentary narrative intelligence within the Soar theory of cognition; we propose to start with narrative intelligence on the most basic of levels, not aspiring beyond child-level narrative intelligence at this point. With this starting point groundwork is laid for future work refining the model.

The implementation proposed proceeds as follows: Soar provides sensory input which is represented in working memory and stored over time as episodes in epmem. These provide the information stream required by EST to make the predictions that result in discrete events. These events are the building blocks of narratives.

3.1 Predictions

At the heart of EST is the making of predictions, which may receive input from a variety of sources including scripts and schema, behavioral character models, genre expectations, and other inputs from semantic memory. As has been previously mentioned the resources available for these processes develops with the experience of the agent. As this exploration considers naive agents with a minimum of prior knowledge it is desirable to have universal heuristics that can form the basis for prediction across domains. Making the simplification that a world consists of agentive and non-agentive components we consider two heuristics. Both of these stand to be superseded as knowledge is gained by the agent.

The heuristic of inertia pertains to non-agentive components of the world, such as spatial configurations. The agent may predict that its environment will continue to exhibit the same features that it now exhibits.

The heuristic of auto-simulation applies to agentive components of the world and takes one of the simplest approaches to a theory of mind by assuming that a perceived agent will act in the same way as the perceiver.

Simplistic as they are, these heuristics provide a ground case to create predictions in any situation, the violation of which delineates the events necessary to form narratives. The result is a stream of events that is, in the worst case of a rapidly and inscrutably changing environment, identical to epmem. With any stability of environment or shared rationality of the agents the product will be an abstraction over the episodes.

3.2 Linking events into narratives

Many definitions of narrative allow for single-event narratives, as when a toddler recalls repeatedly that today “I fell down.” Such interpretation draws no distinction between *event* and *narrative*, a point of ambiguity further promulgated by Zacks’ explanations of EST. The distinction here proposed is not one of structure but of function. EST provides events as a natural kind by which we perceive the world, just as we discern discrete objects. According to EST this perception can occur reflexively. Narrative – particularly personal narrative – is, on the contrary, deliberate and negotiated, the product of an ongoing decision-making process [23] that grows more sophisticated as the narrator matures [4].

Because the aim of this paper is to suggest a means for narrative intelligence that can serve as a (child-like) basis for future work, it is sufficient to allow for single-event narratives while admitting that among the most prominent future work will be the reasoning processes by which more sophisticated narratives can be created from the events produced by EST. These narratives will develop alongside the addition of semantic-memory narrative structures that will influence the top-down processing of EST.

3.3 Considering a Domain: Eaters

While Soar applications are fully capable of recording the richness of real-world perception (e.g. in robotic applications), generating the events with EST which are requisite for narrative generation requires that the system be capable of making useful predictions, which in turn requires rules capturing the complexity of the domain. Games make useful simplified domains. Currently, Soar comes with several game domains that can make testing-grounds for introductory exploration of this approach; we take as an example the Eaters domain [21].

The Eaters game is a two-dimensional Pacman-like game in which one or more colorful "eaters" navigate within a randomly generated maze with the goal of achieving the high score by consuming food pellets of lesser or greater point-values. The eaters are capable of two types of action: moving one space at a time in any of the four cardinal directions, which type of movement has no cost, or jumping up to two squares away, which costs the equivalent of a lesser food pellet. By jumping, an Eater can pass over an obstacle but never consumes food over which it has jumped. When eaters collide, they are each randomly transported elsewhere in the world and their scores are averaged with each other. Each Eater agent has a limited range of vision and discovers the world as it moves. This feature of partial-observability is desirable for mechanisms that rely upon prediction, as does an EST-based approach to narrative intelligence.

3.3.1 Heuristic Prediction in Eaters

Even within so simple a domain as Eaters prediction is still possible and interesting. Because of the partially-observed nature of the domain a natural opportunity for prediction is in world-state itself; for this the heuristic of inertia applies. It happens in Eaters that in randomly generated maps pellets of the same type continue in vertical rows, and that walls may turn but never stagger (do not proceed diagonally or in stair-case formations). The heuristic of inertia means that if the agent has a normal food pellet in front of it as it moves forward, it will predict there to be another food pellet in front after it moves; if not, an event is produced segmenting experience from the previous "normal pellet above" sequence of events. Later reasoning could use this event as a cue to infer that another agent has traversed this path. Likewise, once another Eater has been sighted by an aggressive agent, the heuristic of auto-simulation may come in to play to expect the other Eater to approach. If this doesn't occur, the event might be used in future reflection for the altering of expectations about the unseen portions of the map, or about the schema ("aggressive") of the other agent.

3.3.2 Top-down Narrative Structures in Eaters

A variety of narrative structures could readily be encoded into semantic memory to influence understanding in Eaters. Some such influences could directly influence the production rules applied in Soar by altering the event model being applied. Different event models could include a model for exploration which might apply the afore-mentioned heuristics; prediction error could cue changing to hunting models in which expectations are drawn from heuristics that anticipate perceptual changes that indicate passage of another Eater (e.g. following a trail and expecting pellets to be absent as the trail continues).

3.3.3 Eaters' Narratives

The store of events produced by EST includes segments indicating such things as when a trail of pellets concluded at a wall, or when another eater became visible. In addition to the consideration of these individual events as comprising narratives in their own right, sequences of these events become candidates to be narratives that should be regarded as on a higher hierarchical level than are individual events. Once again the role of top-down structures is important to this production of more complex narratives: as purported by Zacks [35], the changing of event models represents, itself, a key event (e.g. when the agent switches from an exploration model to a hunting model). While the brief model that has been laid out is capable of providing a simple set of event-narratives, these narratives stand to become increasingly interesting and useful as mechanisms for learning semantic structures are introduced.

One of the key features of perception, and hence EST, is the hierarchical nature of perception. Simplified domains like Eaters offer data at a relatively shallow level of abstraction; one way of achieving hierarchical levels of events – and hence higher-level narratives – is by reflection upon episodic memory, by which process broader narrative structures can be applied and recognized. Continuing the Eaters example, reviewing epmem (which contains copies of each state of working memory) can make a place for the application of meta-heuristics, like expecting the heuristic of inertia to apply (say) 70% of the time. This mechanism of heuristics over epmem sequences (rather than singular working memory state) is both naturally precedented by the concept of narrative intelligence, which implies extended temporal breadth, and significant for establishing the recursive nature of narrative.

4 Discussion and Conclusions

The approach to narrative intelligence proposed in this thesis is a preliminary one; it is child-level at best, and awaits further contributions to realize crucial narrative-learning methods that will provide narrative structures, schema, and semantic memory components that are crucial to the next stages of narrative cognition. Such structures proposed by researchers like Propp form the basis of modern narratology and continue to be explored [25, 6, 5]. This model does, however, provide a base-level account for the development of personal narratives from experience. The contribution of this work is to take steps toward a theory of cognitive narrative that bridges the gap between perception and narrative cognition and is, therefore, a comprehensive starting-point for agentive systems. However child-like (even toddler-like) these minimal narratives may be at the start, the function that can provide them will meet needs of both quality and quantity. A system that is able to continually produce narratives from its experiences has the potential to offer the sort of statistical data valuable for categorization and norm detection, both considered some of the fundamental purposes of cognitive narrative in humans [8]. It also offers a promising starting-place for automated generation of scripts within a domain, which could be a useful complement to crowd-sourced script generation that can be costly and unpredictable [18]. Together, these capabilities may serve in support of advanced cognition like goal-based reasoning [30], whereby consideration of narrative schema could provide resources for adaptation or change of goals in dynamic scenarios.

A major question highlighted by the Eaters example with primary relevance to a system's episodic memory has to do with the timing of experiential reflection and personal narrative generation. Although the Eaters example suggests narratives being produced concurrently with perception, much more truthful to work like Ochs' and Capps'[23] is narrative generation that occurs as reflection upon the contents of memory. Indeed, multiple revisits to whatever primitive narratives are produced around perception time will be essential to acquiring higher narrative forms.

Regardless of the episodic memory implementation, a system that produces experiential narratives will also capture qualities of coherence that are desirable in a narrative system. Insofar as narrative is defined as being concerned with having a “continuant subject,” [17] experiential narratives minimally satisfy that by providing the experiencer as subject. This fact is not insignificant for applications in Human-Computer Interactions, Expressive AI, or Affective Computing, where “self” for continuity of subject may provide resources for desirable development of personality and style within an agent [12] and ultimately for the development of life story [27].

An event/prediction-based model of cognitive narrative also extends an invitation to insights from the dramatic arts, whose perspective of narrative as affective is highly relevant to the predictions of EST in response to suspense [24], some of which have already applied Soar [19, 11].

A concluding line of work worth mentioning would be observer-systems which would consider primarily other agents as the subject of their predictions and narratives. Such systems would enhance the quality of the narratives generated by developing narratives based on human or expert-system performance and would be important steps toward tasks such as automated sports commentary [1], summarization [26, 16], and theory of mind [7]. One of the severe challenges facing the development of effective observer systems is having an approach to narrative intelligence that can be generalized across domains. The development of general story-generation algorithms suitable for general cognitive architectures is one strategy for approaching such useful systems; hopefully the approach discussed here is a step in that direction.

Eventually narrative intelligence will be an instrument for general intelligence, at which time we could expect that agents with greater narrative intelligence would have a competitive advantage in games like Eaters. As an introductory exploration, the chief product of the approach proposed are the narratives themselves, preliminary to more advanced functions of intelligence.

References

1 Nicholas D Allen, John R Templon, Patrick Summerhays McNally, Larry Birnbaum, and Kristian J Hammond. Statsmonkey: A data-driven sports narrative writer. In *AAAI Fall Symposium: Computational Models of Narrative*, 2010.

2 Alan Baddeley. The episodic buffer: a new component of working memory? *Trends in Cognitive Sciences*, 4(11):417–423, 2000.

3 David Blair and Tom Meyer. Tools for an interactive virtual cinema. In *Creating Personalities for Synthetic Actors*, pages 83–91. Springer, 1997.

4 Gilbert J. Botvin and Brian Sutton-Smith. The development of structural complexity in children’s fantasy narratives. *Developmental Psychology*, 13(4):377–388, 1977.

5 Mark Alan Finlayson. Learning narrative morphologies from annotated folktales. In *Proceedings of the 1st International Workshop on Automated Motif Discovery in Cultural Heritage and Scientific Communication Texts*, 2010.

6 Pablo Gervás. Propp’s Morphology of the Folk Tale as a Grammar for Generation. In Mark A. Finlayson, Bernhard Fisseni, Benedikt Löwe, and Jan Christoph Meister, editors, *2013 Workshop on Computational Models of Narrative*, volume 32 of *OpenAccess Series in*

Informatics (OASIcs), pages 106–122, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

7 Andrew Gordon. The theory of mind in strategy representations. In *24th Annual Meeting of the Cognitive Science Society*, 2002.

8 David Herman. How stories make us smarter. narrative theory and cognitive semiotics. *Recherches en communication*, 19(19):133–154, 2003.

9 David Herman. Storytelling and the sciences of mind: Cognitive narratology, discursive psychology, and narratives in face-to-face interaction. *Narrative*, 15(3):306–334, 2007.

10 David Herman. *Storytelling and the Sciences of Mind.* MIT Press, 2013.

11 Mikhail Jacob, Alexander Zook, and Brian Magerko. Viewpoints ai: Procedurally representing and reasoning about gestures. In *Proceedings of the 6th Digital Games Research Association Conference, Atlanta, Georgia*, 2013.

12 Nicola King. Memory, narrative, identity. *Remembering the Self Edinburgh University Press, Edinburgh*, 2000.

13 Christopher A. Kurby and Jeffrey M. Zacks. Segmentation in the perception and memory of events. *Trends in Cognitive Sciences*, 12(2):72–79, 2008.

14 John E. Laird. Extending the soar cognitive architecture. *Frontiers in Artificial Intelligence and Applications*, 171:224, 2008.

15 John E. Laird. *The Soar Cognitive Architecture.* MIT Press, Cambridge, MA, USA, 2012.

16 Wendy G Lehnert. Plot units: A narrative summarization strategy. *Strategies for natural language processing*, pages 375–412, 1982.

17 Boyang Li. *Learning Knowledge To Support Domain-Independent Narrative Intelligence.* PhD thesis, Georgia Institute of Technology, 2015.

18 Boyang Li, Stephen Lee-Urban, Darren Scott Appling, and Mark O Riedl. Crowdsourcing narrative intelligence. *Advances in Cognitive Systems*, 2:25–42, 2012.

19 Brian Magerko and John E. Laird. Mediating the tension between plot and interaction. *Ann Arbor*, 1001:48109–2110, 2005.

20 Michael Mateas and Phoebe Sengers. Narrative intelligence. In *Proceedings AAAI Fall Symposium on Narrative Intelligence*, pages 1–10, 1999.

21 Shelley Nason and John E Laird. Soar-rl: Integrating reinforcement learning with soar. *Cognitive Systems Research*, 6(1):51–59, 2005.

22 Allen Newell. *Unified Theories of Cognition (William James Lectures).* Harvard University Press, 1994.

23 Elinor Ochs and Lisa Capps. *Living Narrative: Creating Lives in Everyday Storytelling.* Harvard University Press, 2009.

24 Brian O'Neill. *A computational model of suspense for the augmentation of intelligent story generation.* PhD thesis, Georgia Institute of Technology, 2013.

25 Vladimir Propp. *Morphology of the Folktale*, volume 9. American Folklore Society, 1958.

26 Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408, 2002.

27 G.A. Radvansky and J.M. Zacks. *Event Cognition.* Oxford University Press, 2014.

28 Jeremy R Reynolds, Jeffrey M Zacks, and Todd S Braver. A computational model of event segmentation from perceptual prediction. *Cognitive Science*, 31(4):613–643, 2007.

29 Mark O Riedl and R Michael Young. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research*, 39(1):217–268, 2010.

30 M. Roberts, S. Vattam, D.W. Aha, M. Wilson, T. Apker, and B. Auslander. Iterative goal refinement for robotics. In A. Finzi and A. Orlandini, editors, *Planning and Robotics: Papers from the ICAPS Workshop*, page to appear. Portsmouth, NH: AAAI Press, 2014.

31 Daniel L. Schacter, Donna Rose Addis, and Randy L. Buckner. Remembering the past to imagine the future: the prospective brain. *Nat Rev Neurosci*, 8(9):657–661, 09 2007.

32 Roger C Schank and Robert Abelson. *Scripts, goals, plans, and understanding.* Hillsdale, NJ: Erlbaum, 1977.

33 E. Tulving. *Organization of memory.* Academic Press, New York, NY, 1972.

34 Gillian Wigglesworth. Children's individual approaches to the organization of narrative. *Journal of Child Language*, 24(02):279–309, 1997.

35 Jeffrey M Zacks, Nicole K Speer, Khena M Swallow, Todd S Braver, and Jeremy R Reynolds. Event perception: a mind-brain perspective. *Psychological bulletin*, 133(2):273, 2007.

Optimal Eventfulness of Narratives

Fritz Breithaupt[1], Eleanor Brower[2], and Sarah Whaley[2]

1 Indiana University, Dept. of Germanic Studies
2 Indiana University, Hutton Honors College

Abstract

This study examines whether there is an optimal degree of eventfulness of short narratives. We ask whether there is a specific degree of eventfulness (unexpectedness) that makes them "stick" better than other stories so that they are maintained more faithfully in serial reproduction (telephone games). The result is: probably not. The finding is that there is an impressive correlation of eventfulness rankings of original stories and resulting retellings in serial reproduction, despite the change of many other story elements and almost regardless of low or high eventfulness. Put more simply, people remember and retell "eventfulness" accurately, even when the actual events and circumstances of a story are changed.

1998 ACM Subject Classification I.2.0 General

Keywords and phrases Narrative, Event, Eventfulness, Event cognition, Serial reproduction, Linear and bounded iteration, Event memory

Digital Object Identifier 10.4230/OASIcs.CMN.2015.12

1 Introduction

One of the most central questions of narrative and its cognitive functions is the question of the event. It is hard to imagine narratives without events. There is, however, large disagreement as to what constitutes an event. Are small textual units of actions equal to "events"? Or is an event something larger that occurs in the mind of the recipients who react to a story? In the former case, the event would be a small unit, element, or building block in a sequence of events. In the latter case, events provide the center of gravity that hold all other elements together, like a sun and its planets.

There is certainly space for definitions of events on several levels [6]. Still, in this article we want to explore the second idea that events provide the central point around which entire stories are constructed. However, not every event is able to "tie the knot" equally well. If events have the capacity to tie together larger stories and texts, the question is how one can determine which features make certain events more successful than others in doing so.

To determine the success of narratives, we measure the stability or absence of stability of narratives in conditions of retelling. We define a successfully eventful narrative as narrative that maintains its eventfulness relatively unchanged after retellings.

In this study, we focus on one aspect of eventfulness only, namely its degree of unexpectedness or surprise. Of course, eventfulness encompasses dimensions other than unexpectedness, including relevance, persistence, irreversibility and non-iterativity [13]. Nevertheless, we argue that unexpectedness is a central dimension of eventfulness. In contrast to other aspects of eventfulness, unexpectedness corresponds to a specific experience by recipients. Recipients know when they are surprised, but are less prone to directly experience and report relevance, persistence, irreversibility and non-iterativity, expect in cases when these are strikingly absent. Our study will examine how precisely people reproduce different degrees of unexpectedness when retelling stories.

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 12–22
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

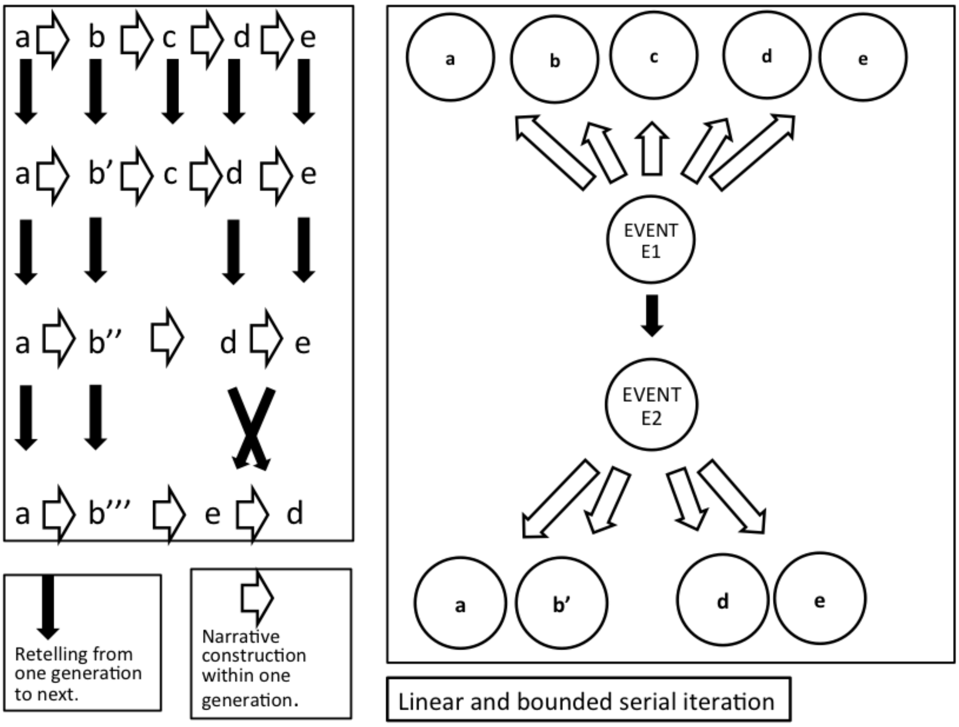

a b c d e
a b' c d e
a b'' d e
a b''' e d
Retelling from one generation to next.
Narrative construction within one generation.
a b c d e
EVENT E1
EVENT E2
a b' d e
Linear and bounded serial iteration

Figure 1 Linear and bounded serial iteration of narratives.

We distinguish two processes or strategies of retelling. In the first process, the story appears as a string of elements with one leading to the next. Retelling means to reconstruct this linear flow of small events from one to the next. Omissions, errors, and transformations occur on the local level, but can affect entire strings that fork off from the original track. In the second process, the narrative is constructed around a core idea. Retelling a story around such a core event means to construct (and invent) all surrounding elements of an event, such as the conditions that lead to the event and the characters. Omissions, errors, and inventions would occur as a consequence of the genetic construction of elements one could expect around the central event. We call these two approaches linear and bounded iterations (Figure 1).

In linear iteration, each element (a, b, c, d, e) would be treated equally and could disappear or change without necessarily affecting the other elements. In bounded iteration, all elements only matter inasmuch as they lead to the constriction of the core event (E1) or can be deduced from the retold event (E2). Elements that are not well connected to the core event are likely to disappear.

It is likely that human retelling uses a combination of both strategies. A one-sided use of linear iteration would likely result in enumerations of seemingly redundant or meaningless elements. A one-sided use of bounded iteration would likely leave out many details and descriptions and thus be impoverished.

In this study, we measure the presence of events and thus bounded iteration after several retellings indirectly by degrees of eventfulness (unexpectedness/surprise). In general, linear and bounded iteration can be measured by means of comparing general survival rate of all story elements on the one hand and those story elements directly related to the events on the other hand. Such a comparison has to take evolutions (changes) of all elements and events into account as well.

A mid-level approach that connects aspects of both strategies can be found in Propp's famous analysis of Russian magic fairytales [10, 3]. Propp's single elements of stories tend to cluster in specific orders or sequences that come closer to bounded narratives. In a similar way Fisseni and Löwe describe super-events that connect sub-events [4].

Logics of linear iteration are somewhat better understood and simpler to describe. However, bounded iteration and the construction of core events is less clearly understood, though much debated [8, 2, 12, 9, 11, 5].

Jerome Bruner articulates the duality between linearity and boundedness of narratives in an elegant way: "What is a narrative? ... A narrative involves a sequence of events. The sequence carries the meaning ... But not every sequence of events is worth recounting. Narrative is discourse, and the prime rule of discourse is that there be a reason for it that distinguishes it from silence. Narrative ... tells about something unexpected, or something that one's auditor has reason to doubt. The "point" of the narrative is to resolve the unexpected, to settle the auditor's doubt, or in some manner to redress or explicate the "imbalance" that prompted the telling of the story in the first place. A story, then, has two sides to it: a sequence of events, and an implied evaluation of the events recounted" (Bruner, 1996: 121) [2].

Bruner does not consider any string of events a narrative, but instead requires that it contain something unexpected or unresolved that focuses our interest. Narratives do not simply list, contain, represent, or express events, but also produce doubt, surprise, suspense, and curiosity in recipients, and this is an essential part of the event, perhaps the event itself.

In this article, we examine whether there is an optimal level of eventfulness that makes a narrative cognitively intelligible, allows for successful recall, and thus permits for coherent retellings. Put simply, is there an optimal level of eventfulness that makes a story stick?

1.1 Optimal eventfulness and serial reproduction

Previously, [9] suggested that stories with minimally counter-intuitive narratives seem to be favored by memory and be cognitively optimal. [9] distinguish intuitive, minimally counterintuitive, and maximally counterintuitive stories on the basis of the mixture of fully intuitive events (corresponding to experience and ontological expectations of the world) and counterintuitive events (not corresponding to experience and ontological expectations of the world). They record how trained experts and a participant group of students rate the number of intuitive and counterintuitive events within a range of well-known and not well-known Grimm fairytales. With this approach they establish there is a sweet spot of just a few but not too many counterintuitive events in those stories that have been culturally most successfully (the best-known Grimm fairytales). These successful stories, it turns out, contain a mix of intuitive and just a few counterintuitive events that mark them as "minimally counterintuitive."

The study by [9] only tangentially deals with issues of story-worlds and genre specific expectations. Fairytales are among the most stylized and culturally coded forms of narrative and may thus be exactly the worst candidate for an examination of narrative in general. It is tricky to imagine how people rate the intuitiveness of events within a fairytale that is clearly marked as a fairytale. Godmothers granting wishes magically to good girls may be quite "intuitive" within fairytales and for people growing up with Disney. However, other participants may mark such a godmother as unlikely and counterintuitive. The forced choice between intuitive and counterintuitive events also may establish, more than anything, the ambiguity of participants having to decide which frame of reference to use: the typical fairytale story-world or the so-called real world.

Nevertheless, the study provides an interesting glimpse into optimal eventfulness of stories. The results by [9] are flanked by a set of studies by Barrett and Nyhof (2001) [1]. Barret and Nyhof used serial reproduction (telephone games) to retell stories. The finding of their studies is that intuitive events that are not bizarre tend to disappear more often than counterintuitive events or intuitive but bizarre events.

Based on [9] and [1], it seems reasonable to speculate that high or midlevel eventfulness is favored for memory and recall in retelling conditions. Hence, we decided to study whether we can establish a more specific level of optimal eventfulness that distinguishes not only between two or three categories, but provides a graded scale.

Accordingly, we established varied levels of eventfulness within the same framing story from very low eventfulness to very high eventfulness. We expected that some of the story versions would survive the retellings better than others and we reasoned that such survival would indicate optimal eventfulness. [9] found that in short-term recall, maximally counterintuitive event sequences were preserved best, while in long-term recall the minimally counterintuitive event sequences were preserved best. Given this distinction between minimally counterintuitive and maximally counterintuitive events, we expected to see some preference for the highly eventful stories since our retelling task was immediate (short-term recall). (We should note again that [9] defined maximally counterintuitive stories as stories with a high concentration of counterintuitive events; as far as we can see, their scale only used a binary distinction between intuitive and counterintuitive single events).

In contrast to these studies, we decided to focus on single-event mini stories. Single-event stories seem better suited to study eventfulness than multiple event stories since multiple event stories may simply cluster events too thickly. Even so, each event may in itself be optimally eventful if it did not stand in too close a proximity to the other events.

We selected stories in which a character is facing a challenging situation. The challenging situation gets resolved by means of events. In this sense, the events serve as connector between challenge and solution. More specifically, the events provide the transition from a state A (challenge) to a state B (solution), from problem to solution, or before and after in line with Hamilton & Breithaupt [5]. Within this story design of an event as connector, eventfulness as surprise can be isolated and formalized by the degree of predictability: The event conforms more or less to typical occurrences within the situation and represents a more or less predictable solution to the challenge. In this story design, the other aspects of eventfulness ([13], see above) are not significant. All events are equally relevant since they solve the challenge (relevance criterion), while persistence, irreversibility, non-iterativity, and genre do not play a strong role due to the brevity of short stories. (An additional aspect of the eventfulness of these stories could be called consistence, as fitting within a single set of event borders [11]).

1.2 Method

1.2.1 Participants

Our participants were found on Amazon Mechanical Turk. We set the Mechanical Turk filter for participants of at least 18 years of age and who were in the United States. Each participant received three different stories of a randomized variation in a randomized order for retelling. Retelling was immediate after each story variation the participant read. Each story branch was retold for three retellings or generations. Each first retelling was routed to just one second reteller and then to a single third reteller. We set filters so that each participant could only participate once in the entire study at any stage.

1.2.2 Materials

We generated a set of three short stories and built seven variations of the key event for each story. These events varied from very minimally eventful (intuitive) to highly eventful (counterintuitive).

The stories were each 3–7 sentences long. Each included a character who found himself or herself in a challenging situation. The opening of the story outlined the situation and the final clause pointed to the solving of the problem or the end of the situation. An example is a "shy" boy who has a crush on a girl, but is too shy to ask her out. Another example is a daughter who has an argument with her mother and runs out of the house into the forest. At the end, the shy boy asks the girl whether she would go on a date with him, and the daughter has built up enough resolve to confront her mother.

For each story, we generated sets of interchangeable middle sentences of varied eventfulness. These middle parts established a transition from the problem or challenge to the ending solution. For example, in the story with the shy boy, we created a range of events that establish how he accidentally meets her under specific circumstances. This could be standing next to her in a line or saving her from a car accident. In pretesting, we asked participants to rank and rate these variations in terms of eventfulness. From the set of variations, we selected seven for each story that in pre-testing appeared to provide a graded variety of eventfulness from very low to very high.

In the basic stories below, XXX marks the part that varies between the versions. The seven versions with a code name (such as "Jason A") and the corresponding severity ranking in brackets (such as "[2.85]") are added behind. The severity rankings given are the median values by participants.

1. Jason liked a girl in his class. He was very shy, however, and was too afraid to talk to her. One day, XXX. He mumbled that she looked nice and asked her if she would like to eat lunch with him.
   - they were standing next to each other in a line (Jason A [2.2])
   - as he was walking down the hallway he saw the girl and noticed that they had on the same outfit (Jason B [2.95])
   - as he was doodling in class, she caught him drawing a perfect likeness of her (Jason C [3.85])
   - as he was walking in front of her desk, he tripped on his shoelaces and fell right in front of her (Jason D [3.85])
   - he decided that to overcome his fear of talking to her he needed to assume an alternate identity. He dressed up as superhero and walked over to where she was sitting (Jason E [5.2])
   - as he was sitting in the classroom, he piled a bunch of different fruits on top of his head and danced over to the girl, while singing her name (Jason F [5.6])
   - as he was walking behind her on the crosswalk to school, he noticed that a car was coming very fast towards them. He quickly ran and pushed her out of the way into safety (Jason G [6])
2. Sarah had a fight with her mother. She ran out of the house. She decided to go into the woods. In the woods, XXX. That made her feel better and gave her the confidence to talk to her mother again. After that, she went back home and apologized.

- she read a book (Sarah A [0.75])
- she stomped around angrily and hit a tree (Sarah B [2.4]).
- she caught a strange looking snake (Sarah C [3.6])
- she dove into the pond and swam around with all her clothes on (Sarah D [4.8])
- she made a fire and burnt everything her mother had ever given her (Sarah E [5.2])
- she found an old racecar that worked and drove it at high speed into a tree (Sarah F [5.6])
- she built a tree house and collected food for a month to stay there (Sarah G [6.1])

3. Robert sat down in class to take his final exam. He knew the exam would be difficult, but he was shocked to see how hard it was. He may not have studied enough, but this exam was simply not fair and he started sweating. With an hour left, he asked for a bathroom break and left the room. In the bathroom, XXX. Then he returned to the testing room to complete the exam.
   - he splashed his face with water (Robert A [0.15])
   - he gave himself a pep talk while washing his hands and loudly sang his favorite song regardless of the other people hearing him (Robert B [2.1])
   - he pulled out his phone and searched the Internet for a couple exam questions (Robert C [3.45])
   - a man he did not know gave him the textbook for his class with all relevant pages for the final marked (Robert D [5.1])
   - he did sprints in front of the stalls to get his brain going. While running, he hit his head on a door, but instead of confusing him, it seemed to cause everything to make sense. (Robert E [5.6])
   - he loudly asked the exam question to the mirror and a voice gave him the answer (Robert F [6.6])
   - he found an envelope with his name on it. Inside was the answer key, signed "with love" from his teacher (Robert G [6.7])

1.2.3 Procedure

We asked participants on Amazon's Mechanical Turk to retell the stories in their own words. We used a variation of instructions from Kashima 2000 [7] that stress that participants should retell stories in their "own words."

The quality of retelling was high. From the selection of retellings discussed in this study, we only disqualified a single retelling on the ground that it was too elaborate (it appeared that the participant wanted to show his or her qualities as writer to embellish a short text into a full page).

Once we received the third retelling, we routed these retellings to (different) participants on Mechanical Turk to evaluate the eventfulness of these stories. Each participant received 20 of the retellings, fully randomized, and was asked to rate the eventfulness on a scale from 0 to 7. We used a slider that also showed the numeric number with one decimal number after the period, such as 5.1. In the instructions, we defined eventfulness as follows:

"A story that is eventful usually contains elements that are surprising or unexpected. In a story that is not eventful, things occur as expected with little or no surprise."

On each screen with a retelling, we also gave the following instructions:

"Please evaluate the eventfulness of the story below from 1–7. 1 would be least eventful; 7 most eventful/surprising. You can use each rating as many times as you feel necessary. If there is no event at all, please mark it as 0."

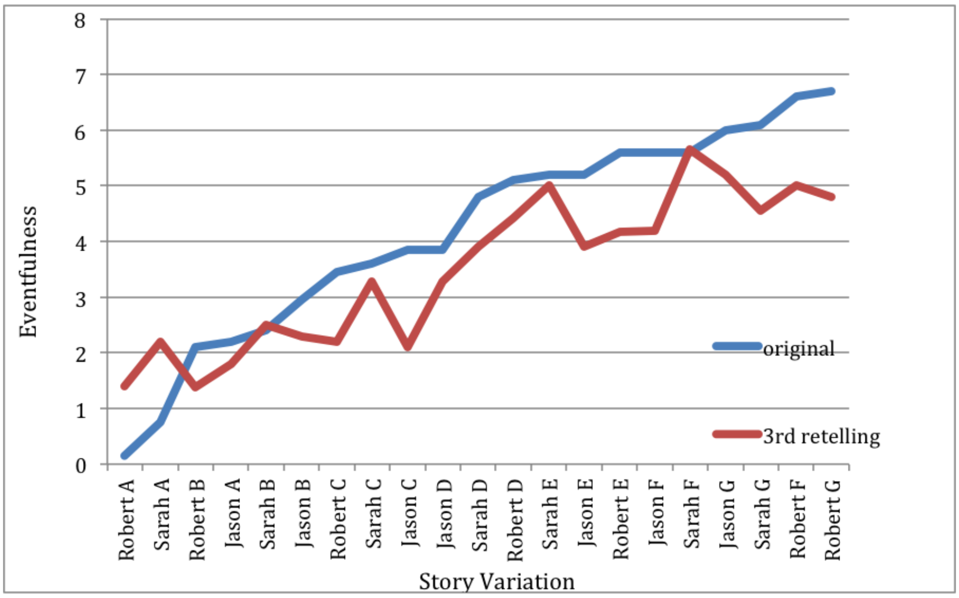


Figure 2 Eventfulness of original and third retellings. The x-axis lists the code names of individual stories. These stories are ordered by eventfulness of the source stories. For example, the source story "Jason C" was rated as less eventful than "Jason D". The y-axis represents the average ratings of eventfulness from 0–7. The chart shows the correlation of source story and the resulting third retelling.

We also used the same approach and instructions to establish the eventfulness of our original or source stories in all variations. Participants who rated the source stories only evaluated source stories in randomized order. Each source story variation received an average of 18 rankings, while the participant retellings received an average of 9 rankings each.

For our calculation of results, we used the median readings of the source stories and compared them with the rankings from the third retellings. For the ranking of the retellings, we established the median value for each individual third retelling and then calculated the median of all individual third-generation retellings that resulted from one story variation. Using the median value is the standard procedure in cases where equidistance between numbers cannot be established. Median values are also less sensitive to outliers than average values, given that a small number of participants may have given random rankings. (Average values, however, returned similar results).

For this present study, we used a set of stories that resulted in a combined 367 third retellings based on the 21 original story variations. That is, the total number of retellings considered here is 1101 (367 first iteration, 367 second iteration, and 367 third iteration). There were between 13 and 24 third generation retellings for each source story (such as "Jason A"). The eventfulness rankings of the third generation stories used a total of 3,375 participant scores.

In the story variations, we decided not to control strictly for length, but instead measure and compare length of different variations. The results of our study focus on eventfulness readings (eventfulness, variance, lengths).

1.3 Results

Three generations of retellings bring about many severe changes in narratives. Typically, the length of stories dropped by around 50%. Much detail disappeared or was radically transformed, as we will indicate below.

Given the wide range of changes, the core finding of this study is even more astonishing. We found a strong correlation between eventfulness rankings of original stories and third retellings, see Figure 2.

Below are the median ranking values of all story variations.

	Jason A	Jason B	Jason C	Jason D	Jason E	Jason F	Jason G
Original	2.2	2.95	3.85	3.85	5.2	5.6	6
3rd retelling	1.8	2.3	2.1	3.275	3.9	4.2	5.2

	Sarah A	Sarah B	Sarah C	Sarah D	Sarah E	Sarah F	Sarah G
Original	0.75	2.4	3.6	4.8	5.2	5.6	6.1
3rd retelling	2.2	2.5	3.275	3.9	5	5.65	4.55

	Robert A	Robert B	Robert C	Robert D	Robert E	Robert F	Robert G
Original	0.15	2.1	3.45	5.1	5.6	6.6	6.7
3rd retelling	1.4	1.375	2.2	4.425	4.175	5	4.8

The results indicate that the eventfulness of a narrative is highly salient for comprehension and retelling, even when many other elements are strongly transformed or dropped at an overall rate of around 50%. The overall correlation coefficient (r) is 0.897659424, thus indicating a strong overall correlation. (Our question of interest is the correlation between source stories and retold versions, hence a non-significant t-test would not allow us to rule out that there is no significant difference).

Furthermore, the results indicate that there is not simply one optimal eventfulness level. Rather, it seems people pay close attention to a given eventfulness level in a story, and preserve and reproduce it accurately, for the most part, even while all other elements are in flux.

The starting hypothesis of a “sweet spot” of optimal eventfulness was not verified. Instead, we noted a strong attentiveness to specific levels of eventfulness.

Only at the extremes of very low and very high eventfulness, below 2 and above 5, do the data suggest a tendency in the retellings to move toward the middle. The ratings of our original stories included extreme ratings of 0.25 and 6.7 for specific stories whereas the ratings after the retelling move closer to the 1.5 to 5.5 eventfulness rating segment.

Based on our original hypothesis, we also speculated that we would find longer lengths of stories to be of an optimal level of eventfulness. This was not the case. The length of third retellings was not correlated with eventfulness, but weakly correlated with the length of the original story, see Figure 4. Correlation values varied for the stories (Jason -0.23; Sarah -0.013; Robert 0.746). The shrinkage was above 50% for the Jason and Robert stories, whereas it was less than 50% for Sarah stories, the shortest original story.

Another predictor we speculated about was story variation. We speculated that some story variations would show a larger variance of eventfulness readings of the individual branches in the third retelling. Again, this was not the case. The variance of eventfulness of individual retelling branches was similar at the extreme ends and the middle ground of the eventfulness scale.

In a future study with more experiments, we will report on full preservation of all elements of the stories. At this point, we should report the high degree of change between original

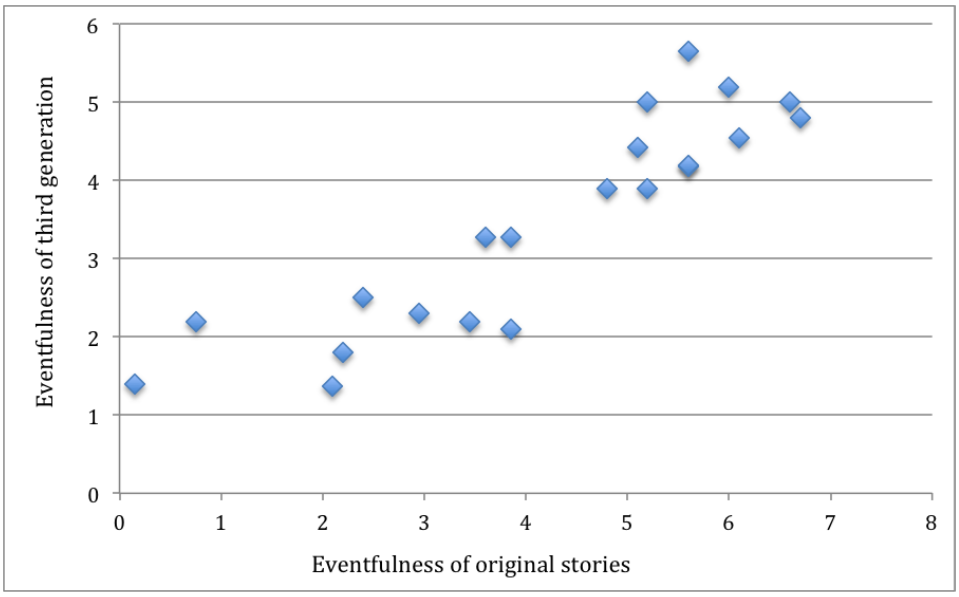


Figure 3 Overall correlation of source stories and 3$^{\text{rd}}$ retelling. The x-axis represents the median eventfulness by the source stories prior to retelling. The y-axis represents the median eventfulness of the third retellings.

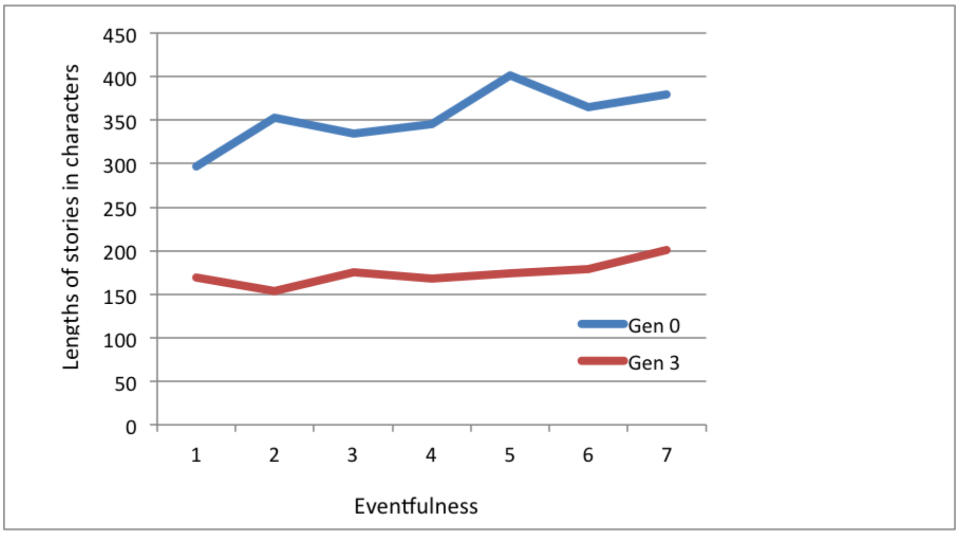


Figure 4 Length correlations between original stories and third retellings, measured in characters.

story and third retelling. As an example, consider one story variation of the shy boy. It started with this text:

"Jason liked a girl in his class. He was very shy, however, and was too afraid to talk to her. One day, as he was sitting in the classroom, he piled a bunch of different fruits on top of his head and danced over to the girl, while singing her name. He mumbled that she looked nice and asked her if she would like to eat lunch with him."

After three retellings, it turned into the following in one of its many branches:

"John fancied a girl in his class. His way to get her attention was to wear a fruit hat and dance his way to her. Mumbling and fumffering, he complimented her appearance and asked for a dance." (J197)

Here, it is interesting to note that the emphasized characteristic of Jason-John as "very shy" disappears, whereas the oddity of his behavior finds a correlate in the neologism "fumffering" (or perhaps from Yiddish funfer, meaning to stutter). Obviously, the original story included the counterintuitive element that a shy boy would do this. Many retellings adjusted this tension by either eliminating the feature of shyness or by dropping details of Jason's odd performance.

This individual string from *shy Jason* to *John the dancer* also illustrates a case in point for the bounded iteration (Figure 1). Linear iteration would have preserved something of the string with the starting proposition (a boy named Jason is shy), the middle action ("one day, . . . he piled fruit on his head. . . ") and the conclusion (he asks her for a lunch date). Instead, the core event around which the retelling is built is the dancing performance of a boy to get the attention of a girl. In classic bounded iteration fashion, other elements are built to fit this middle event, including: he fancied her (beginning) and asked her for a dance (conclusion).

2 Discussion

Our findings suggest that human recipients and retellers of narratives are highly sensitive to specific levels of eventfulness. The specific sensitivity of recognizing and reproducing specific levels of eventfulness accurately allows single-event narratives to maintain eventfulness over multiple generations of retelling. Hence, instead of a single level of optimal eventfulness of narratives, we argue for a broad-range sensitivity of eventfulness of narratives.

Our findings do not dispute that there may be some bias toward some optimal mid-level eventfulness in the cases of multiple events [9, 1]. However, in the condition of single-event retelling, we found much more evidence for an accurate representation of given eventfulness levels. It is possible that the discrepancy of our study and these other studies is a result of changed experimental design. Other studies used multiple-event retellings whereas we focused on single-event retelling. Based on our findings, the more remarkable finding is not the somewhat weaker correlation of very low and very high eventful narratives, but rather the remarkable overall consistency.

Given the impressive degree of correlation between original story eventfulness and third-retelling eventfulness paired with changes of all other story elements, we also suggest that the study supports the idea that narrative retelling makes strong use of bounded iteration. Bounded iteration is a retelling based on the construction of super-events that tie many elements of a given story together. In the process of retelling, the new story is built around and in accordance with the constructed event.

We are currently in the process of validating these findings with different experimental settings and with similar experiments using longer stories. The preliminary findings of the retellings of longer stories are quite encouraging. In the longer stories (12 sentences), the preservation of eventfulness after three retellings is even stronger than in the case of the short stories from this study, while the preservation of the actual events is significantly lower. The preliminary findings strongly support the above finding that eventfulness is better preserved than the actual event.

These findings have significant consequences for generation and comprehension of narratives. They also suggest that we as recipients pay close attention to the eventfulness of narratives. Retelling does not simply preserve semantic or plot-related qualities of narratives, but includes affective dimensions, such as surprise. The degree of eventfulness is linked to expectation and probability. There may be two forces at work here simultaneously that each point in a different direction. One is curiosity. We may constantly look out for something unexpected, unresolved, or surprising. The other force is doubt. When we receive a story, we may constantly monitor its trustworthiness and flag the surprising stories as suspicious. Taken together, this leaves us in a position of having to pay close attention to both the most ordinary account and the most stunning and tall story.

Acknowledgements. We thank the Office of the Vice Provost of Research at Indiana University for an IU Seed Grant to fund the studies presented in this article. We also wish to express our gratitude to Sam Zorowitz who helped us tackle all programming issues, and Torrin Liddell for statistical support, as well as Binyan Li, Zulfukar Ozdogan, and Alan Reiser for their excellent comments.

References

1 J. L. Barrett and M. A. Nyhof. Spreading non-natural concepts: The role of intuitive conceptual structures in memory transmission of cultural materials. *Journal of Cognition and Culture*, 1:69–100, 2001.

2 Jerome Bruner. *Culture of Education.* Harvard UP, Cambridge, Mass, 1996.

3 Mark Alan Finlayson. *Learning Narrative Structure from Annotated Folktales.* PhD thesis, Massachusetts Institute of Technology, 2012.

4 Bernhard Fisseni and Benedikt Löwe. Event mapping for comparing formal frameworks for narratives. *Logique et Analyse*, 57:181–222, 2014.

5 Andrew Hamilton and Fritz Breithaupt. These things called event: Toward a unified narrative theory of events. *Sprache und Datenverarbeitung*, 37:1–2, 2013.

6 Peter Hühn. Event and eventfulness. In Peter Hühn et al., editor, *Handbook of Narratology*, pages 80–97. Walter de Gruyter, Berlin and New York, 2009.

7 Yoshihisa Kashima. Maintaining cultural stereotypes in the serial reproduction of narratives. *Personality and Social Psychology Bulletin*, 26:594–604, 2000.

8 Jurij M Lotman. *The Structure of the Artistic Text.* U of Michigan P., Ann Arbor, 1977.

9 Ara Norenzayan, Scott Atran, Jason Faulkner, and Mark Schaller. Memory and mystery: The cultural selection of minimally counterintuitive narratives. *Cognitive Science*, 30:531–553, 2006.

10 Vladimir Propp. *Morphology of the Folktale.* University of Texas Press, Austin, 1969.

11 Gabriel A. Radvansky and Jeffrey M. Zachs. *Event Recognition.* Oxford UP, Oxford, 2014.

12 Wolf Schmid. Narrativity and eventfulness. In Tom Kindt and Hans-Harald Mülle, editors, *What is Narratology? Questions and Answers Regarding the Status of a Theory*, pages 17–35. de Gruyter, Berlin, 2003.

13 Wolf Schmid. *Narratology. An Introduction.* de Gruyter, Berlin, 2010.

The Evolution of Interpretive Contexts in Stories

Beth Cardier

Sirius Beta Inc.
1976 Munden Point Rd Virginia Beach, VA 23457, USA
bethcardier@sirius-beta.com

Abstract

Modeling the effect of context on interpretation, for the purposes of building intelligent systems, has been a long-standing problem: qualities of logic can restrict accurate contextual interpretation, even when there is only one context to consider. Stories offer a range of structures that could extend formal theories of context, indicating how arrays of inferred contexts are able to knit together, making an ontological reference that is specific to the particular set of circumstances embodied in the tale. This *derived ontology* shifts as the text unfolds, enabling constant revision and the emergence of unexpected meanings. The described approach employs dynamic knowledge representation techniques to model how these structures are built and changed. Two new operators have been designed for this purpose: *governance* and *causal conceptual agents*. As an example, a few lines from the story *Red Riding Hood As a Dictator Would Tell It* are used to demonstrate how a story interpretive framework can be continually re-made, in a way that produces unexpected interpretations of terms.

1998 ACM Subject Classification I.2.7 Natural Language Processing, Discourse

Keywords and phrases Story dynamism, contextual interpretation, ontological interoperability, retroactive revision, narrative progression in discourse processes, derived ontology, situation theory, integrating multiple inferences

Digital Object Identifier 10.4230/OASIcs.CMN.2015.23

1 Narrative and Formal Models of Context

1.1 Introduction

It is difficult for humans to make accurate interpretations across changing contexts, let alone for machines to do so. Bruner observes that for logic, the "world remains invariant" [4, p. 50], and Devlin explains how logical qualities can restrict accurate contextual interpretation, even when there is only one context to consider [11]. This research examines how the structures of stories enable multiple contexts to be managed, proposing two mechanisms (*governance* and *causal conceptual agency*) to account for key aspects of the process. Systematic diagrams represent the formal model [8] and display the mechanisms in animated form [7]. In this paper, a few pivotal frames are provided to indicate their characteristics.

The original aim of this work was to inform the design of a computerized system for intelligence analysis, that captured the way subjective (non-logical) perspectives evolve as they influence each other, rather than how explicit facts add up [6]. Progress has been made towards that system, which is still in development. Its formalisms are not covered here, except to allude to the general mathematical choices made. Instead this paper presents a model of some of the cognitive semantic dynamisms involved in understanding real-world fiction. A companion paper reports on details of the implementation [15].

At the core of this paper are two mechanisms designed for that project: *governance* and *causal conceptual agency*. These operators sit within a description of conceptual integration

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 23–38
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

that is philosophically similar to established approaches in Discourse Processes, such as Kintsch's Construction-Integration Model, in which top-down and bottom-up inferences negotiate [29]. Like that work, this model assumes that the text constrains and informs the memory-based inferences that support reasoning about it. However, this approach departs from previous models in that it is drawn from the issues concerning the composition of compelling fiction. It began with a fiction writer's question: how does a reader anticipate the end of a story she or he cannot predict?

In order to render this artistic concern in the very different field of knowledge representation, a survey of approaches was made, to identify gaps in current models of conceptual structure [8]. Within that domain, the focus was ontological interoperability, which has some known, long-standing problems [40]. One of these issues is directly relevant to the phenomenon of interest: it is difficult to design a system that can automatically bridge incompatible conceptual networks, such as the kind that exist in different knowledge bases. One ontology cannot evolve into another, so that non-logical structures emerge that seem like a natural evolution. I use this problem to frame how stories enable progressive reasoning in ways that differ from current formal models of contextual interpretation.

To clarify this phenomenon, consider the title and first lines of the following story:

> *Red Riding Hood as a Dictator Would Tell It*
> Once upon a time, there was a poor, weak wolf. It was gentle and kindly
> and had a heart of gold [49, p. 230].

Reading from the first phrase, *Red Riding Hood*, to the last phrase *heart of gold*, the reader is led through several different states of expectation regarding themes and events: from a fairytale scenario, to the anticipation of humor and irony mixed with that fairytale scenario (when addition of the dictator is mentioned), and then to the unexpected focus on the wolf with gentle qualities. In order to maintain sense as these expectations shift, some conceptual structures remain stable while others alter. How does this dynamism occur? This paper will outline the way conceptual structure can be built, integrated and revised through mechanisms central to fiction writing.

The resulting model is represented using animations that use conventions of knowledge representation, and extended with approaches such as those of Fauconnier and Turner [13], and Holyoak and Thagard [24] to include dynamism. An animated version of this example can be found online [7]. Figure 1 is a screenshot from this animation, which depicts some of the inferences involved in interpreting the example.

As an introduction, simply notice the bands running across the frame of Figure 2; there are two groups: those at the top, which represent general knowledge structures, and those at the bottom, which represent new, emerging interpretive structure. Connections are woven between them as the text progresses. *Governance*, a new operator, is one of the facilitators of this movement. In Figure 1, a governing node is indicated by the color blue, with lines indicating the direction of effect. *Causal concept agents* are collected in the third situation band from the bottom, fulfilling criteria that will be described in a moment. These new features record the stages of the shift from the general (top) to the specific (bottom), where the new derived ontology is built and changed.

A story's ability to adjust its own frame of reference could offer fresh insight into managing conceptual conflict in systems such as knowledge bases. It could also address the "significant gap" in research on narrative inference identified by Arthur Graesser, who asks "how does the point of a story systematically emerge from the configuration of important goals, actions, obstacles, conflicts, and resolutions expressed in the plot?" [16, p. 239]. This paper proposes that part of the answer can be found in the mechanisms used by a story to handle incompatible

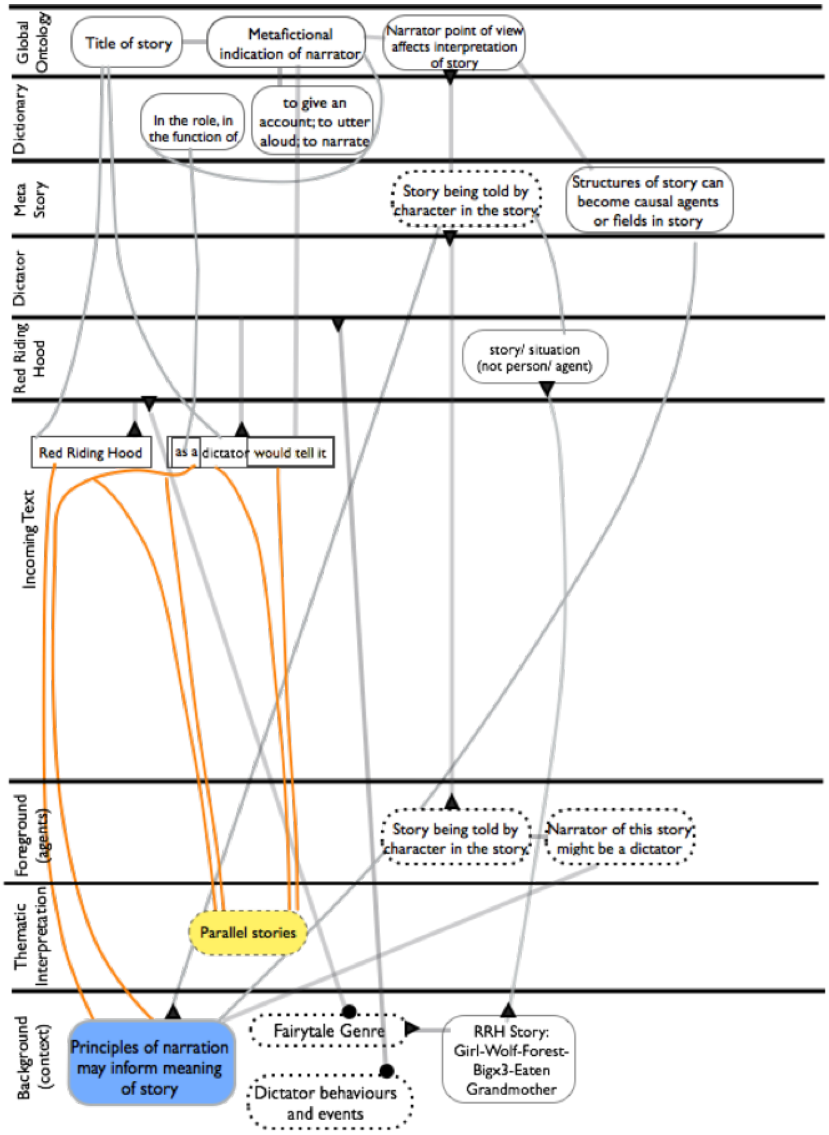

Global Ontology
Dictionary
Meta Story
Dictator
Red Riding Hood
Incoming Text
Foreground (agents)
Thematic Interpretation
Background (context)
Title of story
Metafictional indication of narrator
Narrator point of view affects interpretation of story
In the role, in the function of
to give an account; to utter aloud; to narrate
Story being told by character in the story
Structures of story can become causal agents or fields in story
story/ situation (not person/ agent)
Red Riding Hood
as a dictator would tell it
Story being told by character in the story
Narrator of this story might be a dictator
Parallel stories
Principles of narration may inform meaning of story
Fairytale Genre
Dictator behaviours and events
RRH Story: Girl-Wolf-Forest-Bigx3-Eaten Grandmother

Figure 1 Conceptual structure built by the title of *Red Riding Hood as a Dictator Would Tell It* weaves aspects of general inferences (top) into a new, derived interpretive structure (bottom).

conceptual structures. It will indicate how new referential structure is progressively derived, enabling changes in the interpretation of the terms it supports. Sowa states that a dynamic notion of ontology such as this is needed, to reflect the way the meaning of a word "is unstable and dynamically evolving as it is used in different contexts" [41, p. 245]. This work models some of the structures used by a story to achieve this.

2 Composing the Problem

2.1 Ontology in knowledge bases and stories

The first departure from current literature is the units considered to be fundamental to stories. Formal analyses of narrative often revolve around events and characters in the storyworld

[46, 35, 23], and while these aspects are important, and can be entailed in the abstractions I use, they are not the focus. Instead, this work concerns how stories build and transform the *conceptual structure* used to make inferences during its own interpretation. I refer to this framework as a *derived ontology* [15].

A *derived ontology* is the story's reference framework, one that contains the operating parameters of the story itself, including causal information that enables a reader to understand not only what is happening, but *what can happen.* It includes but goes beyond the notions of *suyet* or *discours* [26], because it entails non-explicit inferences along with the explicit textual devices, and zooms into the granularity of how such structure is built and changed at a conceptual level, so some ideas are deliberately rendered as more important than others. The term *derived ontology* captures these qualities and also indicates fundamental similarities with the computer science notion of ontology. The two instances differ in a few important ways, however.

The term *ontology* was first used in philosophy by Aristotle to refer to the study of being [34, p. 3], and has since been adapted to serve computer science. Here, an ontology is a frame of reference that accounts for a certain view of the world [34, p. 3], and this is also my definition in relation to stories. In both cases, an ontology provides the reference framework used to define terms, similar to a built-in dictionary. It is a "systematic account" of the entities assumed to exist in a domain of interest, as well as the relationships between them [19]. Both stories and knowledge bases can be seen as interpretive machines, in the sense that each relies on an ontology (or something like it) to churn out interpretation. In both stories and knowledge base design, ontology is the reference framework used to make accurate interpretations.

These similarities can lead to confusion regarding the differences. The first distinction concerns generality versus specificity. In computer science, even though an ontology can manifest in a range of different forms [38, p. vi], the common denominator is that it is a static corpus of general reference terms, which have a formal expression [37, p. 61][38, p. vi]. The more this kind of ontology is tailored to a particular domain, the less compatible it will be with those in other systems, a quality termed *heterogeneous* [1, p. 190],[48, p. 164]. In practical terms, this makes a formal ontology similar to a context, because the more specific it is, the more it will be limited to that particular circumstance, and its information less easy to preserve as it is carried to other instances. For this reason, the terms in formal ontologies are chosen to have as "much generality as possible to ensure reusability" [38, p. v]. In this work, systems such as this are thus referred to as a *general ontologies.*

A story does use general references such as this, but then goes further. It draws on numerous general references, and then manipulates elements from them, adding structure until the resulting interpretive framework is unique to the tale. This is a novel contribution of this research: identifying the way that stories construct a new, refined reference situation.

Interestingly, the new derived reference will contain some non-logical structure that does not exist in its sources. To a reader of narrative, these concepts might seem *unexpected* and be less easy to predict [4, p. 12]. There are numerous ways the notion *unexpected* can be defined, it is framed here in relation to paradigms of general assumed knowledge, such as that found in a general ontology. An unexpected conceptual structure is one that is incompatible with commonly known assumption: the sort of structure embodied in a general ontology. The importance of such digression in narrative has been noted across Narratology [23, 3], Discourse Processes [47], and Narrative Psychology [5, 44]. My definition of unexpected includes the way a breach in assumed knowledge can be disruptive, in the manner of Kuhn's "anomaly" which provokes transformation of scientific paradigms [30, p. 6].

Such breach is significant due to the different way systems of logic and story handle anomalous information. In prescriptive logical systems, problems arise when general ontologies encounter unexpected information, and these are so common that a number of approaches have emerged to address them [32]. Most involve some sort of standardisation of terms to eliminate conflict between conceptual structures [38, p. 5]. John Sowa states, “Any incompleteness, distortions, or restrictions in the framework of categories must inevitably omit the generality of every program or database that uses those categories” [40, p. 51]. However, such limits and distortions are an integral aspect of a story’s ability to make sense, and then re-make that sense differently.

Stories can handle unexpected information due to mechanisms that manage the barriers of *context*. A *context* is defined as a limited characterization of reality, which is specific to the peculiarities of a particular circumstance, and contains elements that could not be found easily in other situations. It is information that “is embedded in a specific domain or situation” [39, p. 51], in such a way that information from outside that context might be anomalous. Due to our use of Keith Devlin’s formal system, *Layered Formalism and Zooming (LFZ)* [11], we refer to a context as a *situation* when it takes the form of a discrete conceptual structure. This kind of *situation* has features in common with a heterogeneous ontology, in that its limits can make it difficult to preserve information when it is transferred. In knowledge base design, this can cause problems when different systems try to interact. This is usually addressed through the creation of a large, comprehensive ontology in which all reference frameworks can be situated [32] or the standardization of divergent conceptual structure so that it does not lead to “inconsistent interpretations and uses of knowledge” [20, pp. 381-382]. By contrast, stories leverage such inconsistencies to emulate the flux of the open, real world. Rather than being supported by a single general ontology, or eliminating incompatible ideas, a story’s reference framework enables numerous, limited and diverse conceptual networks to temporarily agree, before changing to accommodate the next chunk of text.

A final area of potential confusion between ontology in the two fields concerns their relationship to logic. In computer-orientated methods, the semantic aspect of the ontology is usually managed by logical rules [40, p. 12], [22, p.30]. In the fictional instance, semantics are structured according to the associative priorities of the story. This structure might contain logical elements, but will also contain many that are not – as Bruner notes, story and logical structures are different modes of thought, “irreducible to one another” [4, p. 11]. When text is interpreted in computer science, the semantic and logical aspects of an ontology are usually the same entity, whereas my model separates them. In the design of a knowledge base, a possible way to handle this would be to build three levels: 1) the semantics of the story ontology, which is structured according to the relations expressed by the story and its reference frameworks; 2) the constructive processes that underpin formation of the story ontology; 3) the logical formalisms that make it computational [15]. Only the first two levels are explored here.

3 Supporting Literature

Modeling contextual inference in unfolding narrative involves several fields, so the supporting literature was drawn from a range of research areas. The following emerged as pertinent: narratological studies on the progressive effects of an unfolding story [44, 27], theories of narrative inference [18, 45, 17], theories of context interpretation and inference [2, 36, 11], current approaches to conceptual integration in knowledge systems [41, 1, 32], and formalisms that concern the representation of narrative conceptual structure [24, 13], as well as their transformation [42, 30]. Of these, a few theories were fundamental to this research.

Foremost was the work of Keith Devlin, whose development of situation theory provided a philosophical foundation and a possible formal framework for its realization. His extension of situation theory, *Layered Formalism and Zooming* (LFZ), is a formal means of expressing the limits of context and the transfer information between them [10]. Devlin's work was extended by our collaborator Goranson to include the narrative properties described here [15]. Devlin's foundations allows for more robust formal methods to be employed in this work.

Discourse Processes was also important, to show how specifics at the perceptive level trigger and restrict generic knowledge inferences [29, p. 125]. Like Kintsch's Construction Integration (CI) model, this work describes continuous conceptual retrieval and adjustment, where only a few nodes actively contribute to the meaning of a node, yet can be easily expanded due to a persistent connection with larger memory structures [28, p. 74]. Although memory and explanation-based processes [21] could both be read into this work, my abstractions are different, so forms of retrieval such as this will manifest and be triggered in relation to different factors. The key difference is ontological conflict; when these models account for contradictions in text [21, p. 244][28, p. 181], they are referring to factual inconsistencies rather than shifts in fundamental definitions of terms. Due to this, and the narrative mechanisms needed to manage it, my expression of these processes differs.

This approach also diverges from Narratology, which usually considers events and characters to be the main features [43, 27, 35, 46]. Michael Toolan examines how text can retroactively attribute importance to particular events, making them cohere in ways that were "unforeseen but foreseeable" [43, p. 215]. In a more formal approach that also focuses on events, Tom Trabasso diagrams the causal dependence of actions in narrative [46, 33], and collaborates with Graesser to consider the forms of inference that produce them [17]. In these cases, the focus on events and activities in the storyworld overlooks a key feature of unfolding narrative: the way the incremental nature of reading can radically change the interpretation of its terms. Cognitive scientist Paul Thagard has argued that further attention to progressive revision is needed to explain "why some revisions are harder to make than others and why some revisions have more global effects" [42, p. 20]. Thagard's diagrams of conceptual change thus provided insights about how contexts evolve [42].

To capture the finer operations of story inference, this approach also draws from Fauconnier and Turner's models of conceptual blending, in which one analogical space supplies conceptual structure, while another is projected into it, making its structures interpretively dominant [13, p. 321]. Fauconnier and Turner do not model the dynamics in the case of an unfolding narrative, however. This means their analogical structure can rest on a fixed general ontology, and the modifications of one situation towards another can be accounted for switching complementary nodes on and off [13, p. 321], rather than the imposition of one structure onto another, so that new structures are formed.

From this survey, several properties of inference in stories emerged as being potentially useful additions to computational models.

4 A Model of Contextual Reinterpretation

Several new mechanisms enable the integration and shift of multiple contexts. Following is an overview of that process, along with a summary of its taxonomic elements.

As a story unfolds, it provokes:

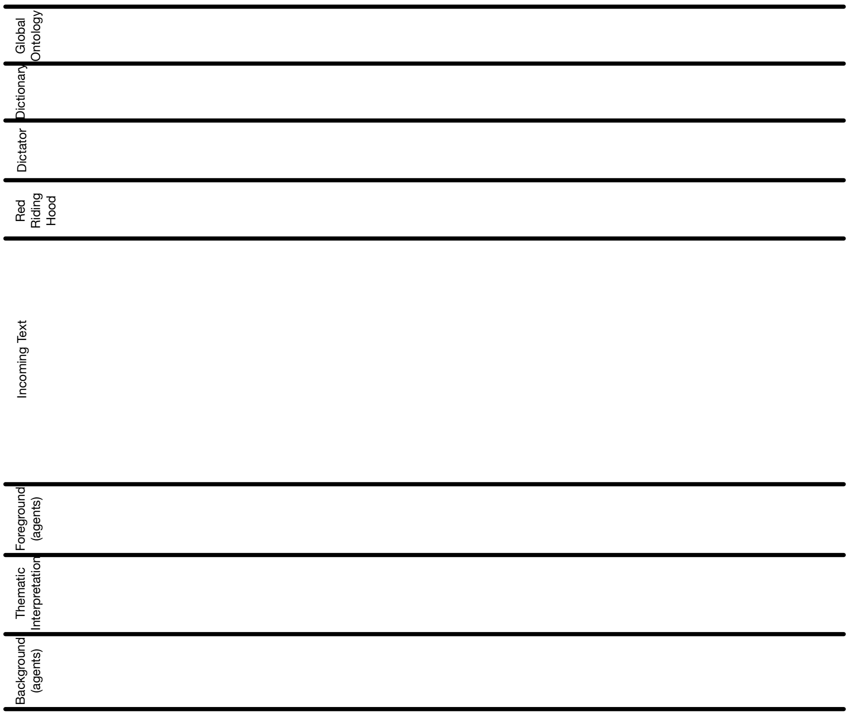


Figure 2 Example of layout, with general and interpretive situations grouped together.

1. **Multiple, limited inferences** which each exhibit properties of context that can make their structures incompatible. These inferences can be connected by

2. **Causal Conceptual Agents**, which contain new structure capable of bridging incompatible inferences. Those new relationships are recorded in a

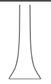
3. **Meta-situation**, in which the ontological structures supporting the various inferences are organized in relation to each other: an ontology of ontologies. This arrangement follows relationships of

4. **Governance**, which enables situations to impose their structures on each other to modify the terms of one network towards another. Altogether, this produces a new reference framework.

Together, these structures form a derived ontology. A summary of the graphical method follows.

In Figure 2, bands are grouped at the top and bottom of the diagram. These are all situations, but the two groups do not perform the same role. Their division represents complementary aspects of interpretation: at the top are situations drawn from general ontologies (the Ontology Space), while at the bottom, the agent network is recorded (the Interpretation Space). The incoming text of the story appears across the middle, so that operators can easily weave structure outwards from it, across the two domains.

The following operators build structure over this framework:

Name / Function	Representation	Graphic
Incoming Text Token	Rectangular box with incoming text inside	Red Riding Hood
Node	Rounded box with concepts inside	Title of story
Link	Line	─
Situation	Encircling box	▭
Pusher	Hammer shape	
Puller	Suction shape	
Funnel	Open V shape	
Dot	Dot	●
Dot (suspended situation)	Dot with white center	○
Wedge	Black triangle, pointing in direction of connection	▼

Of these taxonomic items, the first three (**Incoming Text Token**, **Nodes**, **Links**) are common to conventional methods of knowledge representation. The next three operators (**Situation**, **Pusher**, **Puller**) are new, and capture the behavior of conceptual situations. The first is an encircling box that groups entities to show how their combined structure operates as a single functional unit. The pusher and puller depict the dynamic extraction of subset reference situations.

The **Funnel** instigates change, and as such, is the central structure-building device in this model. In terms of narrative apprehension, it represents an associative connection between actual text and the inferences it provokes. In the graphical depiction, it behaves like a moving arrow, drawing a link between any two objects and creating an attachment between them. Contact with a funnel can change the position and arrangement of concepts, leaving behind an association between the areas of transference. That persistent connection is demonstrated by a grey line. Dots and wedges are superficial indicators that make it easier to decipher the graphical depictions. Dots show where a line starts and ends, like an anchor. Wedges show the direction in which a connection is made, if it is difficult to discern.

There are also eight key states. A *state* indicates what sort of influence a taxonomic element has over its surrounding objects. In order to record the simultaneous development of many elements, states are represented by colors, and can apply to all graphical objects. The colors are not intrinsic to the process being represented, but the differentiation between kinds of activity is important. The states are:

- Neutral (white) □
- Suspended (encircled by a dotted line) ⬚
- Persistent (grey) ▨
- Activation (light yellow) 🟨
- Association-Forming (orange) 🟧
- Conflict (red) 🟥
- Transformative (purple) 🟪
- Governing (blue) 🟦

Neutral (black on white) indicates that the object exists. A dotted black line indicates *suspension*, which means the object *tentatively* exists. A node is registered as tentative when an inference is made that could be salient, but is not yet confirmed (*suspension* is another novel feature). Grey signifies that an object has been built and is now inactive but *persistent.* Yellow signals the *activation* of an existing object. Orange can *associate* objects. Red indicates a *conflict* between associations. At the far end of the spectrum, purple signifies the *resolution* of conflict, while blue indicates *governance.* Both can modify existing structures.

This architecture was used to map the title and first lines of the story *Red Riding Hood as a Dictator Would Tell It* [49] (see above for these lines of text). The story is narrated from the perspective of a sensitive wolf that complains about being persecuted by a girl and her grandmother [49, p. 230]. He explains that one day he wandered into the old lady"s home and was so startled by her that he was forced to eat her. The full story can be found in *The Trials and Tribulations of Little Red Riding Hood* [49]. The animated analysis of these lines can be found online [7].

4.1 Multiple, limited inferences

My example begins when the title *Red Riding Hood as a Dictator Would Tell It* is apprehended. In discourse process models, comprehension begins with a trigger that calls up memory structures [21]; here, such information is drawn from a form of general cultural memory instead. The distinction reflects the phenomenon of interest: part of the skill of professional writing is to judge which inferences can reasonably be assumed of any reader, based on what sort of information is generally known, and what is not. This general knowledge is akin to Arthur Graesser's "generic knowledge structures" [17], and is also similar to the artificial intelligence notion of "common ground"[9, p. 320], where the assumed shared knowledge is the kind a writer can expect of fiction readers they have never met: an example is the kind of information contained in Wikipedia. For ease of reference, that assumed mass audience is referred to as the *reader*, and the shared general cultural memory is collected in the *global ontology.*

In knowledge base design, commonly known examples that might populate the *global ontology* could include *Cyc, WordNet* [40, p. 412] or the coming standard that will enable the *semantic web* [25, pp. 58-59]. Whether for humans, my model, or a computer implementation, this is only the starting point of interpretation, the place from which most foundational reference situations are drawn. Graphically, I depict this collection as a single situation band, running across the top of the frame.

When the first phrase is apprehended, "Red Riding Hood", an inferred cluster of terms associated with the fairytale *Red Riding Hood* is extracted from the *global ontology.* A phrase such as this only activates a limited selection of terms from a general reference framework - this was observed by Kintsch [28, p. 74]. Graesser has referred to a partial inference such as this as a subset of generic knowledge [17, p. 374], and I develop the idea further, to emphasize its properties of context. For example, *Red Riding Hood* is supported by limited conceptual networks regarding the fairytale, and few others. The notion of *dictator* is supported by a few inferences regarding political control and self-aggrandisement. If the supporting ontologies of these terms do not accommodate each other, it might be difficult to relate them on any level. The story will show how they can be linked in this particular circumstance, by adding new structure.

In the graphical example, the extraction of a subset situation occurs when a situation band titled "Red Riding Hood" is pulled out of the global ontology and its dictionary, and rests beneath them, to serve as the first point of reference for further text. The dictionary provides simple dictionary definitions for individual words, whereas the global ontology provides higher-level common knowledge, such as the associations commonly related to the phrase “Red Riding Hood”. The subset titled “Red Riding Hood” is now characterized in terms of the network of terms it contains (I refer to this overall characterization as a *scope*). In this case, the *scope* concerns the fairytale *Red Riding Hood*. The graphical node bears this title, standing in for the terms related to it.

When the term “dictator”, is apprehended, it is tested against the “Red Riding Hood” situation, and no exact match of terms are found. Another subset must be extracted from the *global ontology*, to support it. Finally, with the phrase “would tell it”, a third round of inferencing is provoked. This time, a subset that supports the meta-fictional idea of a “narrator” is extracted. In Figure 1, these subset inferences are depicted as three situation bands, each layered under the next.

When the “Meta Story” situation becomes activated, possible connections become available between the Red Riding Hood and Dictator inferences. Nefarious qualities of the dictator might connect with the role of narrator, after more information is gathered. Perhaps the fairytale plot will feature events from World War II. The focus of this story, both explicitly and implicitly, concerns the bridging of two incompatible situations, but more information is needed to understand how. To confirm which elements will be used and connected, another feature is needed: conceptual agents.

4.2 Causal conceptual agents

Causality is famously difficult to quantify, and the survey of causal philosophy conducted in relation to agency in narrative is covered elsewhere (see [8]). From that literature, Einhorn and Hogarth’s *Judging Probable Cause* was foundational, for the way it describes how causal agency emerges in relation to a contextual field of reference [12, p. 5]. In narrative-related theory, it is common to conceive of agents as characters, and causality as a counterfactual dependence of actions or events (see literature review, above, especially [46]). However, in this work, agency occurs in the context of differing ontological structures. The focus is therefore an aspect of causality more salient to poetics: where causality in story is not a chain of dependence, but a domain of transitions that fit. In this framework, agency is conceptual structure that is able to act on one ontological structure so that it turns into another.

Einhorn and Hogarth’s description of causal agency is embodied in two parameters: *Foreground (causal agents)* and *Background (causal fields)*. These characteristics replaced the single focal situation in Devlin’s formal model of contextual interpretation, *LFZ*, which provided a logical foundation for the formal expression of this work. Graphically, these parameters are represented as horizontal situation bands that run along the bottom of the page (Figure 2). The ***foreground*** band contains nodes that have been identified as conceptual agents, because they exhibit new linking structure. A graphical example in Figure 1, above, would be the node “Narrator might be a dictator”. The central band in this cluster, ***thematic interpretation***, records the most dominant of these, to indicate the overall themes of the story. The bottom-most situation band, ***background***, is composed of nodes that stand in for each inferred reference situation. I refer to these as ambassadors, which will be discussed in the next section.

Agents emerge from the field by virtue of their novel structure (that is, novel compared with what already exists in the reference situations). Their degree of agency is determined by their novelty, as well as how much conceptual structure they are able to link. For example, when the “Meta Story” situation is applied to the whole field, the “Red Riding Hood” and

"Dictator" subsets are cast as separate yet "parallel" situations, ones that will be compared as part of the storytelling. This parallel quality is indicated by the text, with the linking phrase "as a . . . would tell it" but does not exist in any of the subset reference ontologies in isolation. The notion has been derived in relation to their combination. In this case, the node "parallel stories" is an agent because it connects all three subset situations with structure that is novel (compared with what exists in the subset reference situations).

In the implementation, new and transformative structure is informed by Michael Leyton's work on geometric transformation, which illustrates how the evolving topological structures can indicate causal connection [31, p. 3]. When represented as a conceptual network, an ontology endows a story's semantic perspective with structure. When the system searches for structure that will enable transitions between incompatible conceptual structures, it will use semantically-guided topologies to reason about it [14]. Logically, this is expressed as a two-sorted logic, where the second sort uses categoric arrows to reason over situations. This allows semantic-free representation of situations, including those whose explicit facts are unknown.

Causal conceptual agents emerge in relation to the background context being established by the text. In order to examine how that background is composed, let us turn to the *meta-situation*.

4.3 The *Background*: contextualizing contexts

The meta-situation is like an orrery, in the sense that its tokens stand in for a more complex system. Here, in microcosm, relationships between general reference frameworks are built and changed. This miniature is established through gradual honing: general reference frameworks become subsets, which in turn are abstracted as individual nodes, which I refer to as *ambassadors*. Ambassador nodes contain only the most essential elements of the sources from which they were drawn, and are arranged in the meta-situation. Kitsch remarks on the way activated nodes concern only the few elements of general knowledge that are relevant [28, p. 74]; this idea goes further to note how these fragments are positioned in relation to each other by the story. As the text progresses, these tokens are manipulated to reflect the structural priorities of the tale. They carry the relevant aspects of their sources, but have the advantage of being composed of limited conceptual networks, rather than massive general ontologies (although they remain persistently connected to each other), and so are easier to manipulate and modify.

The arrangement of ambassadors, in the form of a meta-situation, serves as an ongoing reference for the incoming text. Agency is relative to a causal field [12, p. 6], and the meta-situation serves as that field. It informs and situates the emerging agents. In implementation, the system will identify nodes as ambassadors for the *Background* situation band if they represent a subset of a reference situation but contain no new structure. Their purpose is to record how the text is building relationships between the reference situations, including which are dominant (dominance will be discussed in a moment). Due to the way the meta-situation shifts as the text progresses, it enables the same word to be interpreted differently as the story unfolds.

Consider the interpretation of "wolf" that would be inferred at different stages of the example story. By itself, the word wolf might be defined as a wild woodland creature with some doglike qualities, and a system using a single ontology would then use this definition as the basis of a composition of facts. In narrative, when the first phrase of the title is parsed, "Red Riding Hood" a quick contextualization occurs: any wolf mentioned at this point would be subject to the terms of the "Red Riding Hood" situation, which would produce the

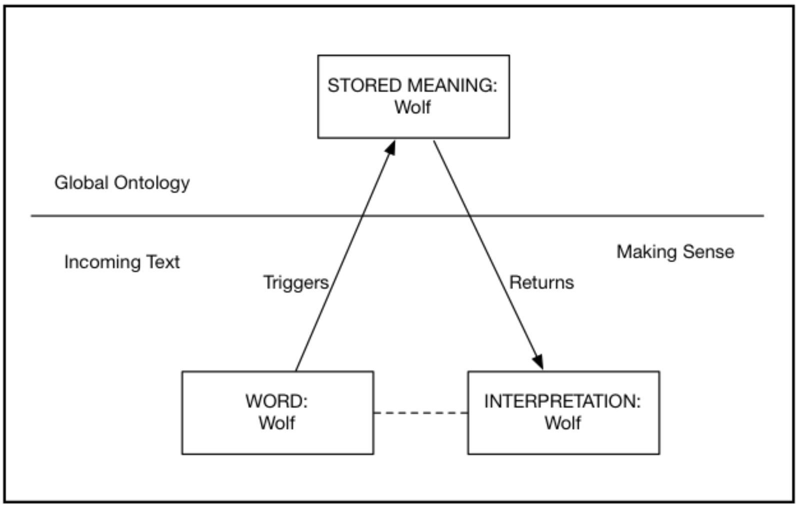


Figure 3 Looking up a word when a single general ontology is the reference.

definition that the wolf is a predatorial character who plans to eat a little girl, perhaps with sexual menace. Below are two illustrations by a collaborator to contrast two different ways "wolf" can be interpreted in this situation [14]. Figure 3 shows the look up when there is a single ontology. Figure shows how the subset situation *Red Riding Hood* could impose its structure to create a more nuanced definition of wolf.

In Figure 3, the definition of 'wolf' is always the same; Figure 4 shows a system in which the terms used to interpret a word can shift with every subset added. The second instance reflects this research, to imitate the way story comprehension can involve many subsets, acting simultaneously.

In *Red Riding Hood as a Dictator Would Tell It*, the nuance does not stop there. The newly defined fairytale 'wolf' is then redefined by the dictator's situation, so that it becomes a character in a story (with predatorial menace) which is of interest to a dictator. By the end of the sentence "It was gentle and kindly and had a heart of gold" [49], the wolf is a dictator, who is narrating the story, and endowed with the dictatorly quality of perverting the truth.

The meta-situation makes co-operation between inferences possible because it records the relationship between them. The variety of means by which this occurs is a large topic of enquiry in itself, and is the subject of ongoing investigation. The basic foundation includes the dynamic that when situations relate to each other, they follow properties of governance.

4.4 Governance

The term *governance* refers to a form of structural imposition. As many inferred situations might compete to have their structures used by the story, a method is needed to designate which take priority; governance fulfills this role. But it is not simply a prioritization method. It also accounts for the adjustments that conceptual structures can perform on each other, modifying conceptual structures so they can connect. In the graphical method, governance is indicated by the color blue (see Figure 1). When one node governs another, the governing node flashes blue and connects to it, and its effect is recorded in the addition or alteration of structure.

Governance can operate at a range of degrees. Its most far-reaching form is demonstrated by the final version of the derived ontology. When a story reaches its end, the final version of

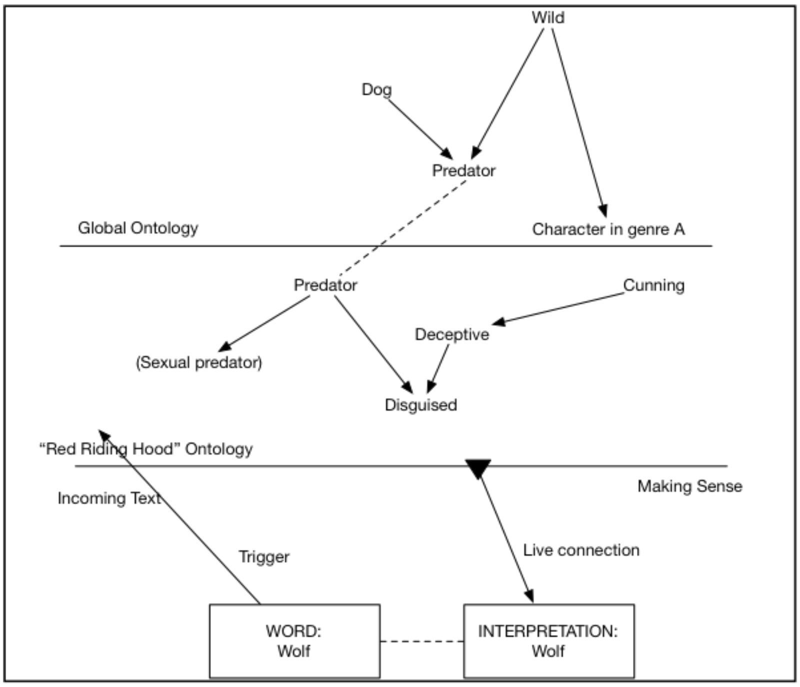


Figure 4 Looking up wolf when each subset reference has different parameters.

the derived ontology acts on the entire tale, retroactively imparting its associative priorities on all previous structures. This can result in major, meaning-altering revisions of the entire network.

In its most local form, governance can act through an individual word, such as the way "wolf" can be considered in relation to the phrase "there was a poor, weak wolf." Here, the words "poor" and "weak" are interpreted on the terms of the governing word, "wolf". Their associative range thus conforms to a scope of qualities appropriate to a fairytale wolf.

Between these two extremes is the most frequently used governance operation. Every time a text chunk appears, a subset situation is used to interpret it. This subset governs the incoming text chunk, in order to provide source structure for that interpretation.

The notion of governance is novel, but is informed by Paul Thagard's research on conceptual change. In *Conceptual Revolutions*, Thagard discusses the transition between two competing theories of combustion, which share the common concept "wood burns" [42, p. 105]. This common node operates as a limited point of attachment between the two incompatible paradigms, and in Thagard's diagrams, acts as a pivot between them.

In narrative, a conceptual agent performs this pivotal role. As the old conceptual framework turns into a new one, the pivot pulls the old structure onto new terms. In a story, there are numerous pivotal points such as this, acting in concert to indicate how one temporarily fixed point can become the next, until the end. Some conceptual structure remain stable while others change. Interpretation can thus evolve and yet comprehension persists, with each temporarily stable point helping to carry the reader to the end.

In a practical sense, governance modifications can occur in numerous ways: one situation might surrender to the associative priorities of the other, or some of its terms might be bent in order to connect to it. The kinds of modification, and under what circumstances they activate, requires further work. More investigation is also required in relation to other aspects of the model: more examples are needed, to explore and refine the taxonomy. In terms of the graphical expression, a richer representation is required for the structure of ambassadors, so it is easier to assess the way they bridge, overlap or conflict with each other. These issues are the subject of ongoing work and collaboration.

In the meantime, this model offers two novel mechanisms towards the issue of bridging incompatible contexts in computable models. It describes how causal conceptual agents use principles of governance to build unexpected conceptual structures. Their dynamic connections thread the narrative transitions together, enabling a reader to track how the themes and central ideas in a story evolve. At each step, the interpretation of the terms of the story alters, as the inferred situations adjust their relationship with each other.

5 Conclusion

This paper presents a novel system to model how narratives manipulate meaning in dynamic and complex ways. Four features of evolving interpretation in stories were identified.

As a tale unfolds, it provokes **multiple inferences** which have properties of contextual limitation. These are connected together by **conceptual agents**, which emerge when different subset situations are applied to incoming text, in such a way that new structure emerges. In order to determine how their differing reference networks should relate, principles of **governance** organize and modify tokens drawn from them. This creates a **meta-situation**, in which tokens of the supporting ontological structures are prioritized and arranged, shifting as the story unfolds. Overall, this constructs a new reference framework, one that is a derivation of the general reference frameworks used, and is specific to a particular set of circumstances embodied by the tale.

These factors combine to give a sense that the interpretative framework of the story is evolving. Narrative mechanisms such as this could offer new insight into problems of interoperability found in knowledge base design. Further study will be pursued to further refine the details of how this process occurs, and shed further light on how an assumed reader is able to anticipate structures they cannot predict.

References

1 G. Acampora, V. Loia, S. Salerno, and A. Vitiello. A hybrid evolutionary approach for solving the ontology alignment problem. *International Journal of Intelligent Systems*, 27:189–216, 2012.

2 B. Britton and A. Graesser. *Models of Understanding*. Psychology Press, New York, NY, 2014.

3 P. Brooks. *Reading the Plot*. Oxford University Press, Oxford, UK, 1984.

4 J. Bruner. *Actual Minds, Possible Worlds*. Harvard University Press, Cambridge, MA, 1986.

5 J. Bruner. The narrative construction of reality. *Critical Inquiry*, pages 1–21, 1991.

6 B. Cardier. Seismic semantic federation: The view from Machu Picchu. In S. Louchart, M. Mehta, and D. L. Roberts, editors, *Intelligent Narrative Technologies II: Papers from the 2009 AAAI Spring Symposium*, pages 12–19, Menlo Park, CA, 2009. Association for the Advancement of Artificial Intelligence.

7 B. Cardier. `https://s3-us-west-2.amazonaws.com/topoiesis.org/Animation+A.mp4`, 2013. [online: accessed April 2015].

8 B. Cardier. Unputdownable. School of Culture and Communication and the Department of Computing and Information Systems. Melbourne, University of Melbourne, 2013.

9 H. H. Clark and T. B. Carlson. Context for comprehension. In J. Long and A. Baddeley, editors, *Attention and performance IX*, pages 313–330. Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.

10 K. Devlin. *Logic and Information*, pages 1–328. Cambridge University Press, Cambridge, MA, 1995.

11 K. Devlin. Modeling real reasoning. In G. Sommaruga, editor, *Formal Theories of Information: From Shannon to Semantic Information Theory and General Concepts of Information*. Springer-Verlag, Berlin, Heidelberg, 2009.

12 H. Einhorn and R. Hogarth. Judging probable cause. *Psychological Bulletin*, 99:3–19, 1986.

13 G. Fauconnier and M. Turner. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books, New York, NY, 2002.

14 H. T. Goranson. Topoiesis document. `http://www.sirius-beta.com/Topoiesis/TopoiesisDocument.html`, 2012. [online: accessed April 2013].

15 H. T. Goranson and B. Cardier. A two-sorted logic for structurally modeling systems. In *Progress in Biophysics & Molecular Biology*, pages 141–178. Elsevier, 2013.

16 A. Graesser, B. Olde, and B. Klettke. How does the mind construct and represent stories? In M. Green, J. Strange, and T. Brock, editors, *Narrative Impact: Social and Cognitive Foundations*, pages 231–263. Lawrence Erlbaum Associates, Mahwah, NJ, 2002.

17 A. Graesser, M. Singer, and T. Trabasso. Constructing inferences during narrative text comprehension. *Psychological Review*, 101(3):371–395, 1994.

18 A. J. Greimas. *Structural Semantics: An Attempt at a Method*. University of Nebraska Press, Lincoln, NE, 1983.

19 T. Gruber. Towards principles for the design of ontologies used for knowledge sharing, Knowledge Systems Laboratory, Stanford University, 1993.

20 M. Gruninger, K. Atefi, and M. Fox. Ontologies to support process integration in enterprise engineering. *Computational & Mathematical Organization Theory*, 6:381–394, 2000.

21 S. Gueraud. Updating situation models: The memory-based contribution. *Discourse Processes*, 39(2-3):243–263, 2005.

22 J. Hendler. Agents and the semantic web. *IEEE Intelligent Systems*, 16:30–37, 2001.

23 D. Herman. *Story Logic*. University of Nebraska Press, Lincoln, NE, 2002.

24 K. Holyoak and P. Thagard. *Mental Leaps: Analogy in Creative Thought*. MIT Press, Cambridge, MA, 1995.

25 I. Horrocks. Ontologies and the semantic web. *Communications of the ACM*, 51:58–67, 2008.

26 P. Hühn, J. Christoph Meister, J. Pier, and W. Schmid (Eds.). *Handbook of Narratology*. Walter de Gruyter GmbH & Co, 2014.

27 K. Ireland. *The Sequential Dynamics of Narrative*. Associated Uni. Presses, Plainsboro, NJ, 2001.

28 W. Kintsch. *Comprehension: A Paradigm for Cognition*. Cambridge University Press, Cambridge, UK, 1979.

29 W. Kintsch. An overview of top-down and bottom-up effects in comprehension: The ci perspective. *Discourse Processes*, 39(2-3):125–128, 2005.

30 T. S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago, IL, 1973.

31 M. Leyton. *Symmetry, Causality, Mind*. MIT Press, Cambridge, MA, 1992.

32 L. Li and Y. Yang. Agent-based ontology mapping and integration: Towards interoperability. *Expert Systems*, 28:197–220, 2008.

33 D. McNamara and J. Magliano. Toward a comprehensive model of comprehension. In Brian Ross, editor, *The Psychology of Learning and Motivation, Vol. 51*, pages 297–384. Academic Press, Burlington, MA, 2009.

34 A. Pretorious. Introduction and overview, semantic technology and applications research laboratory. `https://www.starlab.vub.ac.be/teaching/Ontologies_Intr_Overv.pdf`, 2004. [Online: accessed August 2013].

35 M.-L. Ryan. *Possible Worlds, Artificial Intelligence and Narrative Theory.* Indiana University Press, Bloomington, IN, 1991.

36 R. Schank and R. Abelson. *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1977.

37 B. Smith, W. Kusnierczyk, D. Schober, and W. Ceuters. Towards a reference terminology for ontology research and development in the biomedical domain. *Biomedical Ontology in Action*, 222:57–65, 2006.

38 B. Smith and C. Welty. Ontology: Towards a new synthesis. In C. Welty and B. Smith, editors, *Formal Ontology in Information Systems*, pages iii–x. ACM Press, Ongunquit, ME, 2001.

39 J. Son and R. Goldstone. Contextualization in perspective. *Cognition and Instruction*, 27:51–89, 2009.

40 J. Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations.* Brooks/Cole, Pacific Grove, CA, 2000.

41 J. Sowa. The role of logic and ontology in language and reasoning. In R. Poli and J. Seibt, editors, *Theory and Applications of Ontology: Philosophical Perspectives.* Springer, Berlin, Germany, 2010.

42 P. Thagard. *Conceptual Revolutions.* Princeton University Press, Princeton, NJ, 1992.

43 M. Toolan. Graded expectations: On the textual and structural shaping of readers' narrative experience. In J. Pier, editor, *The Dynamics of Narrative Form*, pages 215–238. Walter de Gruyter, Berlin, New York, 2004.

44 M. Toolan. *Narrative Progression in the Short Story.* John Benjamins Publishing Company, Philadelphia, PA, 2009.

45 T. Trabasso. Goal plans of action and inferences during comprehension of narratives. *Discourse Processes*, 39(2-3):129–164, 2005.

46 T. Trabasso and L. Sperry. Causal relatedness and importance of story events. *Journal of Memory and Language*, 24:595–611, 1985.

47 T. van Dijk and W. Kintsch. *Strategies of Discourse Comprehension.* Academic Press, New York, NY, 1983.

48 P. Visser, D. Jones, T. Bench-Capo, and M. Shave. An analysis of ontology mismatches; heterogeneity versus interoperability. In *AAAI 1997, Spring Symposium on Ontological Engineering*, pages 164–172, Palo Alto, CA, 1997. Association for the Advancement of Artificial Intelligence, Stanford University.

49 J. Zipes (Ed.). *The Trials and Tribulations of Little Red Riding Hood.* Routledge, New York, NY, 1993.

Structured Narratives as a Framework for Journalism: A Work in Progress

David A. Caswell

Structured Stories
Los Angeles, California, USA
david@structuredstories.com

Abstract

This paper describes Structured Stories, a platform for producing and consuming journalism as structured narratives based on instantiations of event frames. The event frames are defined using FrameNet and are instantiated as structured events using references to nodes in various knowledge graphs. Structured narratives with recursive, fractal and network characteristics are then assembled from these structured events. The approach requires the direct reporting of journalistic events into structure by untrained reporters, and utilizes a simplified sequential user interface to achieve this. A prototype has been built and published, and is being applied to the reporting of local government journalism to explore editorial aspects of the approach.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods, F.4.1 Frames and Scripts

Keywords and phrases journalism, frames, narrative structure, event representation

Digital Object Identifier 10.4230/OASIcs.CMN.2015.39

1 Introduction

Journalism has historically been produced, distributed and consumed within the context of loosely-defined supra-document products such as edited newspapers and magazines. These products provide not merely collections of discrete text articles, but also larger-scale informal narrative functions across articles, such as story continuity, consistency of voice, de-duplication, indicators of importance, variance in detail, loose organization of sub-narratives, etc. They are often perceived by their producers and by their consumers to be conveyers of coherent supra-document narratives [3].

More recently, due to technological disruption, the economic basis of these products has started to break down, or 'unbundle', and they are increasingly being replaced by digital streams of isolated text documents, often clustered and ranked using topic models and named entity references. This unbundling has had negative consequences for professional journalism producers, for whom the economic and competitive advantages of supra-document journalism products have been replaced by intense article-to-article competition. It has also had some negative consequences for journalism consumers, who have gained access to far greater quantities of text articles but who have simultaneously lost the large-scale organizing and narrative functions that supra-document journalism products provided.

Computational models of narrative may offer an alternative form of supra-document journalism product that could resolve some of the consequences of unbundling for producers and consumers of journalism, and that may be sustainable in the current economic and technological environment. Considerable work has been performed on this, most often focused on extracting structured storylines from vast corpora of text articles using supervised and semi-supervised natural language processing techniques that are trained on small sets of documents

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 39–44
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

carefully annotated using various annotation schemes – an approach that is exemplified by the ongoing EU NewsReader project [12]. These automated story understanding systems must directly confront the complexity of natural language, albeit via machine learning, and remain dependent on sources of high-quality natural language text articles that are under severe and increasing economic threat.

Alternative approaches that provide mechanisms for the direct creation and maintenance of structured narratives as journalistic artifacts have not been widely explored in recent years, perhaps because the structures used by earlier direct-entry narrative modeling systems, such as the scripts of Ableson and Schank [1] and even the sketchy scripts of DeJong [5], have been formal, complex and therefore difficult to apply in a production journalism environment. The more recent availability of new networked knowledge management technologies does not appear to have been applied to new attempts at direct-entry narrative modeling beyond a few examples such the BBC storyline ontology [11] and Facebook's custom stories [9].

Structured Stories is an attempt to build and test a platform for supra-document journalism products using event and narrative data structures. The approach does not attempt a formal representation of events and narratives equivalent to that expressible in natural language, but instead provides a 'computational pidgin' for narrative somewhat similar to that proposed by Margaret Masterman and Martin Kay for machine translation in 1960 [10]. Events within Structured Stories are considered to be discrete things in the world, in the Davidson sense [4], and not linguistic artifacts originating in text. The arrangement of these events into narrative structures seeks to align with human narrative cognition concerning the relative importance of events and the encapsulation of detail within narratives.

The Structured Stories platform was designed and built during late 2013 and 2014, and has been implemented as a cloud-hosted and API-accessible database of event and narrative information. It is currently being populated with structured narratives in the local government domain, and is consumable in five languages.

2 Description of the Platform

The building blocks of Structured Stories are event frames, which are abstractions of discrete journalistic events and are defined as subsets of FrameNet semantic frames [2]. Event frames are light-weight and flexible and are gathered into a searchable library that can grow to many tens of thousands of frames. Each event frame contains a set of type-constrained event roles that are referenced to semantic roles within the parent semantic frame, and a set of natural language phrases that are centered on a verb lexical unit from the semantic frame and that express event-level context. Although rooted in the semantic formalism of FrameNet, these contextual phrases characterize event frames as editorial artifacts, and not as formal structures. As editorial artifacts they are therefore relatively simple and flexible, and are intended to be created, managed and used by journalists for journalistic purposes.

Listing 1 Event frame – simplified structure for a 2-role event frame

```
Event frame ID
FrameNet frame ID
Role1 ( Event Frame Role , FrameNet Role , allowed type )
Role 2 ( Event Frame Role , FrameNet Role , allowed type )
Phrase 1 ( Journalistic Phrase , Verb Lexical Unit  )
```

Discrete journalistic events are represented within the platform as structured events. Each structured event is defined by an event frame, and each of the event roles from the defining event frame is assigned a typed reference to a Uniform Resource Identifier (URI) – typically an entry in a knowledge graph. These URIs are constrained by type and the platform recognizes seven top-level types: characters, entities, locations, information artifacts, other events, narratives and constants. The knowledge graphs used include Freebase, WikiData and Facebook, and the event type and narrative type are referenced to structured events and structured narratives within the Structured Stories database. Structured events are also associated with various discourse elements, including natural language bullet points, summaries describing the event, images illustrating the event, etc., and events are also linked by cause and effect relationships.

Listing 2 Structured Event – simplified structure

```
Event ID
Event frame ID
Time ( reference time, temporal granularity, temporal duration )
Role references:
   Characters ( event frame roles, knowledge graph IDs )
   Entities/concepts (event frame roles, knowledge graph IDs)
   Locations (event frame roles, knowledge graph IDs)
   Information artifacts (event frame roles, local references)
   Reference Events (event frame roles, event IDs)
   Referenced Stories (event frame roles, story IDs)
   Constants (event frame roles, local references)
Discourse elements (text summary, image, audio, video, etc.)
Causal relationships (causing event IDs, cause types)
```

The platform represents narrative structures as ordered collections of references to structured events, with each reference carrying information about the function of the event within the structured narrative. The relative importance of the event within the structured narrative is represented, and the encapsulation of detail about the event is captured using references to other structured narratives. This fractal-like [6] and recursive structuring enables single structured narratives of many tens of thousands of discrete events to be represented coherently and explored with a few clicks. The narrative structure also enables linkages between structured narratives using common events, common characters, common locations and several other factors, enabling very large-scale narrative networks to be assembled and navigated.

Listing 3 Structured Narrative – simplified structure

```
Story ID
Story events
        ( Event ID, Importance Value, Subnarrative Story ID )
        ( Event ID, Importance Value, Subnarrative Story ID )
        ( Event ID, Importance Value, Subnarrative Story ID )
                                ...
```

These event and narrative structures enable an array of features that facilitate the consumption of journalism. The presentation of narratives can be extensively controlled, enabling the use of different kinds of discourse elements to provide different media experiences of the narrative. The use of structured narratives appears to substantially improve the consumption efficiency of narratives compared with consumption from documents by providing explicit control of detail, access to sub-narratives and navigation of the narrative network. Source documents and related documents are linked from individual structured events and are therefore easily findable within the narrative structure. Text discourse elements can be translated at the event level using machine translation or single-sentence human translation

– a much easier task than the translation of large multi-sentence narratives within text documents. The basis of structured narratives as a permanent and cumulative data store enables the publishing of journalism as a 'pull' (user decides) model rather than as a 'push' (publisher decides) model. Individual events are published as URIs and might therefore be used in mashups or in fact-checking applications, and explicit querying of the event and story database using knowledge graph references, semantic frame references and other structural elements is straightforward. Even reasoning on structured narratives may be possible.

The technical implementation of the prototype platform is centered on a RESTful API, powered by a Node.JS server application. The databases are hosted on Amazon AWS EC2 and S3, and combine Redis, a file system and a graph database. The front-end application is based on the AngularJS application framework.

3 Discussion

Significant uncertainty exists regarding the ease with which untrained users can create and edit structured events and structured narratives within the platform, and also regarding their motivation to do so. Exploring this uncertainty is one of the primary goals for the project and has driven the design of several features within the platform.

The approach seeks to provide sufficient expressive power in its representation of events and narratives to be useful for journalism, but simultaneously seeks to be simple enough to enable easy use by untrained users – typically professional and citizen journalists. This 'goldilocks' goal has been addressed through the light-weight and flexible nature of the event frames, and through a sequential user interface technique that has been shown to enable the entry of individual events by an untrained reporter within 20 seconds.

The approach seeks to deliberately manage the risk of combinatorial explosion in the number of event frames in multiple ways. There is a deep design assumption that the distribution of the use of event frames for journalism will follow a scale-free power law [7], and therefore that the combination of a library of 'head' event frames, a fast method for creating new 'tail' event frames, and a fast search mechanism for finding event frames will enable frame numbers to be manageable. The risks of combinatorial explosion in editorial tasks, such as event frame de-duplication, are higher but are partly reduced by the use of FrameNet as a semantic foundation.

The near-term challenge of motivating participation by reporters during experimentation will be initially addressed by employing a small number of reporters to add structured events and assemble structured narratives in small domains with strong journalistic needs – specifically local government journalism in selected cities. In the medium term motivation will likely depend on the prospect of a sustainable economic rebundling of journalism as structured narrative products and on civic motivation by citizen journalists. In the long term motivating participation by reporters would depend on the efficacy of structured narratives as a mechanism for accumulating journalism and for distributing that journalism via novel products. There are also many additional significant uncertainties regarding the utility of the approach to consumers of journalism, upon which the motivation for participation by producers will ultimately depend.

4 Next Steps

The prototype of the Structured Stories platform is currently being populated with structured events and structured narratives relating to local government news stories in Los Angeles.

The next step for the project will focus on evaluating the feasibility of event and narrative entry and maintenance by untrained reporters, and on defining and evaluating editorial processes to facilitate the management of journalistic quality within structured narratives. This evaluation will occur concurrently with a major reporting project focused on local government in New York City, which will be undertaken during the summer of 2015. If reporting and editing prove feasible then a deep evaluation of the consumption side of the approach, using the captured structured narratives and an iOS app, will be attempted.

Regardless of the results of this testing the Structured Stories project should generate a dataset of hand-curated journalistic news events referenced to FrameNet frames and semantic roles, populated by knowledge graph references and linked to text articles that describe those news events. This dataset may be useful as a training set for supervised machine learning projects. Conversely, there are opportunities to use machine learning techniques such a relation extraction and frame parsing to facilitate capture of structured events into the platform. The Structured Stories approach to modeling narrative structure is therefore an alternative to, and also a complement to, the supervised machine learning approach.

Several extensions to the Structured Stories platform are anticipated, and include the addition of sources of event semantics beyond FrameNet (including VerbNet, PropBank and possibly the NewsReader Events and Situations Ontology), the inclusion of additional discourse elements at the structured event level (including audio, video and comics), and the possible extension of discourse elements to individual roles within the structured events. Improvements to the event reporting workflow, possibly including semi-automation of the workflow using the EVITA system [8] and various TF-IDF document clustering techniques such as the Associated Press Overview system will be explored following the assessment of reporting and editing using the prototype platform.

The Structured Stories prototype is publicly available at `http://www.structuredstories.org`.

References

1 Robert Abelson and Roger Schank. *Scripts, Plans, Goals and Understanding.* Psychology Press, USA, 1977.

2 Collin Baker. Framenet, present and future. In *First International Conference on Global Interoperability for Language Resources*, 2008.

3 Allan Bell. *The Language of News Media.* Wiley-Blackwell, New Jersey, USA, 1991.

4 Donald Davidson. The Logical Form of Action Sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action.* University of Pittsburg Press, 1967.

5 Gerald DeJong. Prediction and Substantiation: A New Approach to Natural Language Processing. *Cognitive Science*, 3:251–273, 1979.

6 Charlie Hargood et al. Exploring the Poetics of Strange and Fractal Hypertexts. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media.* 23rd ACM Conference on Hypertext and Social Media, 2012.

7 Lev Muchnik et al. Origins of power law degree distribution in the heterogeneity of human activity in social networks. *Scientific Reports*, 3, 2013.

8 Roser Saurí et al. EVITA: A Robust Event Recognizer for QA Systems. In *Proceedings of HLT/EMNLP 2005*, pages 700–707, 2005.

9 Facebook Inc. Open Graph Stories (`https://developers.facebook.com/docs/opengraph`). *Facebook Developer Website*, 2015.

10 Margaret Masterman and Martin Kay. Mechanical Pidgin Translation (`http://mt-archive.info/CLRU-1960-Masterman.pdf`). *Cambridge Language Research Unit*, 1960.

11 Robin Pembrook. Storyline Data Model: Sharing the Ontology for BBC News (`http://www.bbc.co.uk/blogs/internet/entries/98d9d3f2-6932-371b-931c-7a13fbf1bacf`). *BBC Internet Blog*, 2013.

12 P. Vossen, G. Rigau, L. Serafini, P. Stouten, F. Irving, and W. R. V. Hage. Newsreader: recording history from daily news streams. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC'14)*, Reykjavik, Iceland, 2014.

Impulse: A Formal Characterization of Story[*]

Markus Eger, Camille Barot, and R. Michael Young

Liquid Narrative Group
North Carolina State University, USA
{meger,cbarot,rmyoung}@ncsu.edu

Abstract

We present a novel representation of narratives at the story level called Impulse. It combines a temporal representation of a story's actions and events with a representation of the mental models of the story's characters into a cohesive, logic-based language. We show the expressiveness of this approach by encoding a story fragment, and compare it to other formal story representations in terms of representational dimensions. We also acknowledge the computational complexity of our approach and argue that a restricted subset still provides a high degree of expressive power.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Narrative, logic, representation, mental models, time

Digital Object Identifier 10.4230/OASIcs.CMN.2015.45

1 Introduction

Narrative is used across cultures to convey both fictional and non-fictional stories. This ubiquity has led to narrative research in many fields, from narrative theory to linguistics to cognitive psychology to AI. Within AI, research ranges from understanding and reasoning about existing narratives to generating new ones. In this field, the division narratologists make between *story* and *discourse* is often used [3]. The story consists of the events that happen in the story world while the discourse describes how these events are told. For example, a story may consist of a murder, an investigation and an arrest, in that order, but a movie rendition may start with the investigation and end with a flashback to the murder to "reveal" the murderer, i.e. the order the events are shown differs from the order in which they actually happened.

We propose a representation for the story level of a narrative called Impulse. In addition to the representation of core story elements such as events and actors, it also provides means to encode information that is not essential to the story but may be relevant for reasoning about possible discourses. Furthermore, Impulse allows complex reasoning about the story itself. We will show how this reasoning can be used to derive explanations for characters' actions or beliefs. We claim that Impulse provides a strong basis for building systems to computationally reason over stories, for story understanding, analysis, as well as for discourse generation.

[*] This material is based upon work supported in whole or in part with funding from the Laboratory for Analytic Sciences (LAS). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 45–53
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Related Work

Due to the wide variety of research interests of scholars building computational models of story, there is also a variety of representations, each highlighting different aspects of a story. Elson and McKeown [5] describe a system for encoding stories in graphs, designed to allow structural comparison between different narratives. A tool allows for easy encoding, annotation and comparison of stories, but it lacks rich formal inference rules.

Some story generation systems also produce stories in a representation that is suitable for further processing. For example, partial-order causal link planning with intentions (IPOCL) has been described as a generative approach for stories by Riedl and Young [13], as an improvement over their previous work with POCL plans [12]. An IPOCL plan consists of steps, that are linked to other steps with causal and temporal links, and frames of commitment that represent character intentions. The model of time in the plan is necessarily simple, to keep the planning process computationally feasible. Furthermore, there is no representation for character beliefs. Very closely related to planning is Martens *et al.*'s [9] use of Linear Logic to generate stories, but their representation does not include time or actors' mental models either.

Ontologies are also often used to represent stories, for example in the Drammar model [8]. Drammar provides an operationalization of a Belief, Desire, Intention (BDI) model represented as an ontology. Swartjes and Theune [14] have elaborated on an earlier version of this ontology by incorporating Trabasso *et al.*'s General Transition Network [16]. However, these approaches only consider relative ordering of steps. Swartjes and Theune also reiterate the point made by Tuffield *et al.* [17] that formal characterization of story generation systems' outputs is still lacking. In particular, when the story is to be presented to an audience by a discourse generator, representing exact timing information is crucial. The discourse generator Darshak, for example, uses a representation of time, based on the planning algorithm DPOCLT, for precisely that reason [7]. When using external data sources, such as video games, precise timing information is available, but if this knowledge can not be represented, it would be lost and could not be reasoned about.

Allen and Ferguson's representation of actions and events in interval temporal logic (ITL) allows complex reasoning over time [2], and remedies shortcomings of the situation calculus [10], like the frame problem. It is based on predicate logic, uses intervals as its representation of time, and includes actions as first-class objects. The representation already allows rich reasoning about the story content and deduction of new facts, but does not contain any model of the actors' mental models. On the other hand, Cohen and Levesque's [4] BDI model, which is also based on predicate logic, allow the representation of, and reasoning about, actors' mental models that would allow inferences about characters' motivations, but does not include a representation of time. We present a novel representation of narratives at the story level, called Impulse, that combines ITL with a BDI model to improve upon the limitations of these representations.

3 Representation

Impulse is based on ITL, a representation based on predicate logic, and augments it with a BDI model of actors. We will first describe the temporal representation we use and how it can be reasoned about. Then we will discuss how time can be added to predicate logic, and how to represent actions and objects in a story, closely following ITL. We then discuss the integration of BDI models with this temporal representation.

Table 1 Allen's interval relations and their representation in Impulse.

Name	Allen	Definition	Notation
Equal	$t_1 = t_2$	$\mathsf{start}(t_1) = \mathsf{start}(t_2) \wedge \mathsf{end}(t_1) = \mathsf{end}(t_2)$	$t_1 = t_2$
Before	$t_1 < t_2$	$\mathsf{end}(t_1) < \mathsf{start}(t_2)$	$t_1 \prec t_2$
Meets	$t_1 \mathrm{~m~} t_2$	$\mathsf{end}(t_1) = \mathsf{start}(t_2)$	$t_1 : t_2$
During	$t_1 \mathrm{~d~} t_2$	$\mathsf{start}(t_1) > \mathsf{start}(t_2) \wedge \mathsf{end}(t_1) < \mathsf{end}(t_2)$	$t_1 \sqsubset t_2$
Starts	$t_1 \mathrm{~s~} t_2$	$\mathsf{start}(t_1) = \mathsf{start}(t_2) \wedge \mathsf{end}(t_1) < \mathsf{end}(t_2)$	$t_1 \lhd t_2$
Finishes	$t_1 \mathrm{~f~} t_2$	$\mathsf{start}(t_1) > \mathsf{start}(t_2) \wedge \mathsf{end}(t_1) = \mathsf{end}(t_2)$	$t_1 \blacktriangleright t_2$
Overlaps	$t_1 \mathrm{~o~} t_2$	$\mathsf{start}(t_1) < \mathsf{start}(t_2) < \mathsf{end}(t_1) < \mathsf{end}(t_2)$	$t_1 \leadsto t_2$

3.1 Representation of time

Impulse uses intervals as its unit of time. Conceptually, an interval t is a non-empty "stretch" of time, with a start and an end, denoted by $\mathsf{start}(t)$ and $\mathsf{end}(t)$, respectively. We will denote the set of all possible intervals with T, called the *time basis*. Two intervals can be in one of 13 different relations to one another, called Allen's interval relations [1]. Table 1 gives an overview of 7 of them with the notation used in Impulse, where the missing 6 are simply the inverses of all but the equality relation.

▶ **Definition 1.** Multiple basic interval relations can be combined into a set $\{R_1, \dots, R_n\}$, where each of the R_i is one of Allen's 13 interval relations. Then $t_1\{R_1, \dots, R_n\}t_2 \Leftrightarrow t_1 R_1 t_2 \vee \dots \vee t_1 R_n t_2$.

One important complex relation is the subinterval relation:

▶ **Definition 2.** An interval t_1 is a *subinterval* of an interval t_2, written $t_1 \sqsubseteq t_2$, iff the two intervals are the same, or t_1 is during, starts or finishes t_2, i.e. $t_1 \sqsubseteq t_2 \Leftrightarrow t_1\{\sqsubset, =, \lhd, \blacktriangleright\}t_2$.

3.2 Temporal and atemporal predicates and functions

To make the step from predicate logic to one based on time, predicates and functions can now have an additional "time" parameter over which they hold. We call predicates and functions with this parameter *temporal* and those without *atemporal*. For example $at(John, Library, t)$ means "John was at the Library for the interval t", and at is a temporal predicate. We use the same concepts of strong and weak negation as Allen and Ferguson:

▶ **Definition 3.** The *strong* negation of a temporal predicate P over an interval t, written $\neg P(p_1, \dots, p_n, t)$ states that the predicate is false during any subinterval of t, i.e.

$$\neg P(p_1, \dots, p_n, t) \Leftrightarrow \neg \exists t_1 \in T\; t_1 \sqsubseteq t \wedge P(p_1, \dots, p_n, t_1)\,.$$

▶ **Definition 4.** The *weak* negation of a temporal predicate P over an interval t, written $\sim P(p_1, \dots, p_n, t)$ states that the predicate is false during some subinterval of t, i.e.

$$\sim P(p_1, \dots, p_n, t) \Leftrightarrow \neg \forall t_1 \in T\; t_1 \sqsubseteq t \rightarrow P(p_1, \dots, p_n, t_1)\,.$$

Furthermore, we require all predicates used in Impulse formulas to be homogeneous.

▶ **Definition 5.** A predicate is called *homogeneous* iff it being true over some interval t implies that it is also true over every subinterval of t, i.e.

$$\forall t_1 \in T\; P(p_1, \dots, p_n, t) \wedge t_1 \sqsubseteq t \rightarrow P(p_1, \dots, p_n, t_1)\,.$$

Temporal functions present another challenge, as they may change value over time, leading to situations where their value may be undefined, i.e. functions are partial with respect to time. For example, if $f(t_1) = a$ and $f(t_2) = b$, the value of $f(t_3)$, with $t_1 \sqsubseteq t_3 \wedge t_2 \sqsubseteq t_3$, is undefined. Using an undefined value in any way will propagate that value, and any predicate on an undefined parameter does not hold.

3.3 Representation of objects and actions

Objects in Impulse are objects in the predicate logic sense, representing concrete and abstract entities in the story world and being uniquely identified by name. All objects in the story are collected in a set O, of which arbitrary subsets can be defined to be used by formulas. Two of these subsets, $A \subseteq O$ and $L \subseteq O$, represent the actors and locations in the story respectively, and have to be defined for all stories. These subsets provide a "type system" for the objects, allowing sentences to refer to objects of specific types. For example, a sentence could say that all locations are cold, without saying anything about other objects.

Similar to objects, actions are elements of a set called *Actions*, with a subset defined for each different action type. For example, there could be a *move*-action set, which is a subset of *Actions*, containing all possible *move*-actions. Normally, we will not be concerned with all possible actions, but only with those that actually happened or could have happened in a particular story. What determines the uniqueness of each action are its properties:

▶ **Definition 6.** A *property* p of an action type $Y \subseteq \mathit{Actions}$ is an atemporal function $p : Y \mapsto O$.

For example, an action of type *openDoor* may have a property door : *openDoor* ↦ *Doors* that refers to the door being opened by a specific action of the action type *openDoor*. Additionally, properties of temporal values are also supported:

▶ **Definition 7.** A *time interval property* q of an action type $Y \subseteq \mathit{Actions}$ is a function $q : Y \mapsto T$.

To distinguish between actions that actually happens in the story and those that are only part of the reasoning process of some character, a predicate occurs is introduced.

▶ **Definition 8.** The atemporal predicate occurs(e) holds if and only if e is an action that actually happens in the story.

An action will typically have some predicates associated with it that have to hold for the action to be possible, and other predicates that describe the effect of the execution of that action. Like ITL, Impulse uses Skolem functions called pre$n$ and eff$n$ on actions to describe the duration of their preconditions and effects. Suppose we have an action "open the door", then its effect can be encoded as $\forall s \in \mathit{openDoor}\ \exists t_1, t_2\ \mathrm{occurs}(s) \wedge \mathrm{closed}(\mathrm{door}(s), t_1) \rightarrow \mathrm{open}(\mathrm{door}(s), t_2)$. However, this leaves us with the existentially quantified variables t_1 and t_2 that depend on the story, i.e. when the *openDoor* action happens, and when the door was previously closed. Allen and Ferguson argue that the sentence $\forall s \in \mathit{openDoor}\ \mathrm{occurs}(s) \wedge \mathrm{closed}(\mathrm{door}(s), \mathrm{pre1}(s)) \rightarrow \mathrm{open}(\mathrm{door}(s), \mathrm{eff1}(s))$ is equivalent to the preceding encoding, but now the intervals depend on the action instantiation directly, and we can now also refer to them in formulas.

3.4 Actors' mental models

Impulse uses a simplified representation of actors' mental models, in the form of a BDI representation. This has previously been used for narrative representation [11]. It allows us to represent character beliefs, which are important to reason about disparity between their views of the world, and - when used with a discourse realizer - with the audiences view of the world as well as their desires and intentions which are important to reason about how to deduce and convey character motivations. While this model does not capture every aspect of character's mental models (e.g., emotional state), we argue that a limitation of the representation is essential to allow inferences to be made in a reasonable manner, and that a BDI model provides sufficient details to reason about a story for discourse generation. It is also possible to extend this mental model representation for specific applications, or to represent emotional states as predicates in the existing Impulse formalism.

Because of our representation of time, the modal operators for belief, desire and intention had to be modified to include a temporal parameter as well:

▶ **Definition 9.** $B_a(t)\Phi, D_a(t)\Phi$ and $I_a(t)\Phi$, with $a \in A$ an actor, t a time interval over S and Φ an arbitrary Impulse formula represents that actor a *believes, desires* or *intents* the formula Φ, respectively.

Note that the temporal parameter actually belongs to the modal operator. Φ will contain its own temporal information. This allows us to represent complex relations like "From 8AM to 10AM John believed that dinner would be served from 7PM to 8PM, but then someone told him that it was actually served from 6PM to 7PM, so he revised his belief".

The only property Impulse enforces on beliefs, desires and intentions is homogeneity:

▶ **Definition 10.** Beliefs, Desires and Intentions are homogeneous, with respect to time, i.e. $\forall t\, \forall t_1\ (B_a/D_a/I_a(t)\Phi \wedge t_1 \sqsubseteq t) \Rightarrow\ B_a/D_a/I_a(t_1)\Phi$.

Other properties often encountered in BDI models can be defined as needed. For example, one may want to define that beliefs are always consistent:

▶ **Definition 11.** $\forall t : B_a(t)\Phi \Rightarrow \neg B_a(t)\neg\Phi$, for any Impulse formula Φ.

3.5 Story representation

A complete story consists of:

- a time basis T, which is a set of intervals,
- an object hierarchy, with O the set of all objects and a definition of subsets thereof,
- an action hierarchy, with *Actions* the set of all actions and a definition of subsets thereof,
- a set of action properties P, as functions mapping from actions to objects or intervals,
- a set of actions Σ that occur in the story. This means $s \in \Sigma \Leftrightarrow \text{occurs}(s)$,
- a set of Impulse sentences Ψ

With this representation, a deduction system can reason about the story by applying logical operations on the sentences in Ψ and deriving new facts. Alternatively, an explanation system could remove steps from Σ or add new ones and then reason about "what would have happened". A discourse generation system, on the other hand, can reason about which information has to be presented to the audience, and which one can be deduced. Depending on what should be conveyed, it may also decide to show or not show the duration of actions.

4 Evaluation

4.1 Example

The example presented here is a shortened version of a scene from the movie "The Lord of the Rings: The Fellowship of the Ring", based on the book of the same name [15]. In the movie, Isildur, the king of men, comes into possession of a magical ring. One of his allies, the elf Elrond, knowing that the Ring is "evil", advises him to destroy it, but the Ring has too much influence over its bearer. In the movie, this leads Elrond to conclude that men are weak. For space reasons, we omit many of the movie's actions and only present the most important ones.

As a time basis, we use intervals over the natural numbers, so $T \subseteq \mathbb{N} \times \mathbb{N}$, and denote "the interval starting at (and including) a and ending at (and not including) b" with ${}_at_b$. The objects in the story include *Elrond*, *Isildur* and *Ring*, so $O = \{\mathit{Elrond}, \mathit{Isildur}, \mathit{Ring}, \mathit{Aragorn}, \acute{E}owyn, \ldots\}$, the set of actors is $A = \{\mathit{Elrond}, \mathit{Isildur}, \mathit{Ring}, \mathit{Aragorn}, \acute{E}owyn\} \subseteq O$ and the set of locations $L = \{\} \subseteq O$. We also define a set $Humanoid = \{\mathit{Elrond}, \mathit{Isildur}, \mathit{Aragorn}, \acute{E}owyn\}$ used to prevent the Ring from actively doing anything, and a set $men = \{\mathit{Isildur}, \mathit{Aragorn}, \acute{E}owyn, \ldots\}$ containing all the human actors[1]. The Ring plays a special role in the story, so the function bearer(t) is used to keep track of who is the Ring-bearer at any given time. We have three action types:

- *get* represents an actor getting the Ring. It has the associated property $\mathrm{actor} : \mathit{get} \mapsto \mathit{Humanoid}$, and a single effect duration $\mathrm{eff1} : \mathit{get} \mapsto T$
- *tellToDestroy* represents an actor telling another one to destroy the Ring. It has the properties $\mathrm{actor} : \mathit{tellToDestroy} \mapsto \mathit{Humanoid}$, $\mathrm{recipient} : \mathit{tellToDestroy} \mapsto A$, one precondition duration $\mathrm{pre1} : \mathit{tellToDestroy} \mapsto T$ and two effect durations: $\mathrm{eff1}, \mathrm{eff2} : \mathit{tellToDestroy} \mapsto T$
- *succumb* represents an actor succumbing to the will of the ring, it has one property $\mathrm{actor} : \mathit{succumb} \mapsto \mathit{Humanoid}$ and two effect durations $\mathrm{eff1}, \mathrm{eff2} : \mathit{succumb} \mapsto T$

Note how *tellToDestroy* can only be performed by a *Humanoid*, but the recipient may be any actor. So, in theory, an actor could tell the Ring to destroy itself. These actions don't actually "do" anything, though, so we need to define what happens when they occur in a story:

1. $\forall s \in \mathit{get}\ \mathrm{occurs}(s) \rightarrow \mathrm{bearer}(\mathrm{eff1}(s)) = \mathrm{actor}(s)$
2. $\forall s \in \mathit{tellToDestroy}\ \mathrm{occurs}(s) \wedge \mathrm{allies}(\mathrm{actor}(s), \mathrm{recipient}(s), \mathrm{pre1}(s)) \rightarrow D_{\mathrm{recipient}(s)}(\mathrm{eff1}(s))\,\mathrm{destroyed}(\mathit{Ring}, \mathrm{eff2}(s))$
3. $\forall s \in \mathit{succumb}\ \mathrm{occurs}(s) \wedge \mathrm{bearer}(\mathrm{pre1}(s)) = \mathrm{actor}(s) \rightarrow I_{\mathrm{actor}(s)}(\mathrm{eff1}(s)) \neg\,\mathrm{destroyed}(\mathit{Ring}, \mathrm{eff2}(s))$

The other Impulse sentences representing the story are:

4. $\mathrm{allies}(\mathit{Isildur}, \mathit{Elrond}, {}_1t_{10})$
5. $\forall t \in T\ \forall a, b \in A\ \mathrm{allies}(a, b, t) \rightarrow \mathrm{allies}(b, a, t)$
6. $\forall t\ D_{\mathit{Ring}}(t) \neg\,\mathrm{destroyed}(\mathit{Ring}, t)$
7. $\forall t\ D_{\mathit{Elrond}}(t)\,\mathrm{destroyed}(\mathit{Ring}, t)$
8. $\forall t \in T\ B_{\mathit{Elrond}}(t)\,\mathrm{weak}(\mathit{Isildur}, t) \rightarrow \forall m \in \mathit{men}\ B_{\mathit{Elrond}}(t)\,\mathrm{weak}(m, t)$
9. $\forall t \in T\ D_{\mathit{Ring}}(t)\Phi \rightarrow D_{\mathrm{bearer}(t)}(t)\Phi$
10. $\forall t \in T\ t_1 \in T, \forall a \in A,\ D_a(t)\Phi \wedge D_a(t)\neg\Phi \wedge D_{\mathit{Elrond}}(t)\Phi \wedge I_a(t)\neg\Phi \wedge t : t_1 \rightarrow B_{\mathit{Elrond}}(t_1)\,\mathrm{weak}(a, t_1)$

All these sentences form the set Ψ. Additionally, we have to state which actions actually occur in the story, and the values of their properties, i.e. the contents of Σ:

- $s_1 \in \mathit{get}$ with $\mathrm{actor}(s_1) = \mathit{Isildur}$, $\mathrm{time}(s_1) = {}_1t_2$, $\mathrm{eff1}(s_1) = {}_2t_5$
- $s_2 \in \mathit{tellToDestroy}$ with $\mathrm{actor}(s_2) = \mathit{Elrond}$, $\mathrm{time}(s_2) = {}_2t_3$, $\mathrm{recipient}(s_2) = \mathit{Isildur}$, $\mathrm{pre1}(s_2) = {}_1t_2$, $\mathrm{eff1}(s_2) = {}_3t_5$
- $s_3 \in \mathit{succumb}$ with $\mathrm{actor}(s_3) = \mathit{Isildur}$, $\mathrm{time}(s_3) = {}_3t_4$, $\mathrm{pre1}(s_3) = {}_2t_3$, $\mathrm{eff1}(s_3) = {}_4t_5$, $\mathrm{eff2}(s_3) = {}_4t_{10}$

[1] As in the movie, we use "men" to refer to "the race of men", i.e. humans, rather than "males".

Table 2 Comparison of the expressiveness of Impulse and other story representations.

Story aspect	IPOCL	ITL	BDI	SIG	Drammar	Impulse
Temporal representation	Limited[a]	Rich	None	Limited[a]	None	Rich
Beliefs	None	None	Rich	Rich	Rich	Rich
Desires	None	None	Rich	Rich	Rich	Rich
Intentions	Limited [b]	None	Rich	Limited[c]	Rich	Rich
Alternate timelines	None	Rich[d]	None	Rich	None	Rich[d]
Formal semantics	Rich	Rich	Rich	Limited[e]	Rich	Rich

[a] Relative order and instantaneous steps; DPOCLT has durations but only simple interval relations
[b] Intentions are used to justify why actions are taken, but no further reasoning is done on them
[c] Story Intention Graphs only have "goals", and no strong distinction between "desires" and "intentions"
[d] Alternate/imagined timelines can be represented by sequences of actions that did not occur
[e] Story Intention Graphs allow comparison of stories, but there are no formal inference rules

Together, the time interval, object hierarchy, action hierarchy, action properties, sentences and occurring actions form the "story". We can now derive additional information about it:

11. allies$(\mathit{Elrond}, \mathit{Isildur}, {}_1t_2)$ (from 4 and 5, and homogeneity of predicates)
12. bearer$({}_2t_5) = \mathit{Isildur}$ (from 1 and $s_1 \in \mathit{get}$)
13. $D_{\mathit{Isildur}}({}_3t_5)$ destroyed$(\mathit{Ring}, {}_3t_5)$ (from 2, 11 and $s_2 \in \mathit{tellToDestroy}$)
14. $D_{\mathit{Isildur}}({}_3t_5)\neg$ destroyed$(\mathit{Ring}, {}_3t_5)$ (from 6, 9 and 12)
15. $I_{\mathit{Isildur}}({}_4t_5)\neg$ destroyed$(\mathit{Ring}, {}_4t_{10})$ (from 3, 12 and $s_3 \in \mathit{succumb}$)
16. $B_{\mathit{Elrond}}({}_4t_{10})$ weak$(\mathit{Isildur}, {}_4t_{10})$ (from 7, 10, 13, 14, 15 and homogeneity of desire)
17. $\forall m \in \mathit{men}\ B_{\mathit{Elrond}}({}_4t_{10})$ weak$(m, {}_4t_{10})$ (from 8 and 15)

We thus conclude that Elrond believes men to be weak. In the movie, this is conveyed as a flashback. With Impulse, a discourse generator could reason about the story to generate such a scene, or a story authoring tool could be used to explore what changes would prevent this belief from forming, e.g. an alternative story in which Elrond believes in the strength of men.

4.2 Expressive power

As the example above demonstrates, Impulse allows for rich reasoning about facts in the story and the mental models of the actors. Table 2 shows a comparison between Impulse and other story representations discussed in section 2 in terms of which aspects of the story they can represent. As can be seen in this table, other representations are more limited in their representation of time or actors' mental models when compared to Impulse.

4.3 Usage

The expressive power of Impulse comes with a price: computational complexity and even decidability. Since Impulse is an extension of predicate logic, which is already undecidable in the general case [18] and computationally expensive in many others, using it as-is is not feasible. However, just like Horn clauses [6] are a subset of predicate logic that allows a more efficient reasoning process while still providing expressiveness, subsets of Impulse can be identified for similar uses. We propose to limit all sentences to two forms:

- *Facts* are single predicates without any connectives, but with optional quantifiers, e.g. $\forall t\ D_{\mathit{Ring}}(t)\neg$ destroyed(Ring, t)

- *Rules*[2] consist of a single implication, where both the antecedent and the consequent consisted of “and”-connected facts, also with quantifiers, e.g.
  $\forall t \in T\, \forall a, b \in A\ \text{allies}(a, b, t) \rightarrow \text{allies}(b, a, t)$

Limiting the sentences to these two forms allows us to use a slightly modified variant of forward chaining, that accounts for the temporal aspect of the logic, as a more efficient method for deriving new information. As the Lord of the Rings example demonstrates, these two forms are sufficient to represent and reason about a complex narrative.

Since Impulse is designed for story representation rather than for generation, data must be acquired and encoded in Impulse somehow. There are several ways this can happen. One approach is to use a story encoded in another representation, for example as an IPOCL plan, and translate it to Impulse. Then this story could be annotated manually or automatically to make use of Impulse’s richer representation of time and actors’ mental models, for example by using a scheduler, or doing intention recognition. Another rich data source for content describable in Impulse are log files of video games. They often contain very detailed information about the states of the world and which actions are performed by actors over time, as well as having detailed and formal rules for the effects of their actions. A discourse generator could use this information to provide e.g. a summary of the game in an engaging way.

5 Conclusion

We presented Impulse, an expressive logical representation for stories that incorporates representations of time and actors’ mental models of the world. It draws from Allen and Ferguson’s work on Interval Temporal Logic and combines it with a BDI model, which is modified to also account for time. We demonstrated how this approach can be used to model a simple story fragment and reason about its actors’ mental models. We then compared the expressive power of our representation to that of other approaches. We also acknowledged the computational complexity of the reasoning process on our representation, and how it can be limited for some particular use cases. We argue that one such restriction yields an efficient, yet expressive deduction scheme. An actual implementation of this deduction system is currently being worked on.

While we claim that this representation could be used in a discourse generator, a tighter integration and a representation of the discourse itself still remains as future work.

References

1 James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

2 James F Allen and George Ferguson. Actions and events in interval temporal logic. *Journal of logic and computation*, 4(5):531–579, 1994.

3 Seymour Benjamin Chatman. *Story and discourse: Narrative structure in fiction and film.* Cornell University Press, 1980.

4 Philip R Cohen and Hector J Levesque. Intention is choice with commitment. *Artificial intelligence*, 42(2):213–261, 1990.

[2] Rules often correspond to definitions of what happens when an action occurs. The terms in the antecedent and consequent are thus called respectively “preconditions” and “effects”, which explains the naming of the pre$n$ and eff$n$ functions.

5 David K Elson and Kathleen R McKeown. A platform for symbolically encoding human narratives. In *Proc. of the AAAI Fall Symposium on Intelligent Narrative Technologies*, 2007.

6 Alfred Horn. On sentences which are true of direct unions of algebras. *The Journal of Symbolic Logic*, 16(01):14–21, 1951.

7 Arnav Jhala and Robert Michael Young. Cinematic visual discourse: Representation, generation, and evaluation. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2):69–81, 2010.

8 Vincenzo Lombardo, Cristina Battaglino, Antonio Pizzo, Rossana Damiano, and Antonio Lieto. Coupling conceptual modeling and rules for the annotation of dramatic media. *Semantic Web Journal, Special Issue on Reasoning and Rule Systems*, 2015.

9 Chris Martens, Joao F Ferreira, Anne-Gwenn Bosser, and Marc Cavazza. Generative story worlds as linear logic programs. In *Proc. of the Seventh Intelligent Narrative Technologies Workshop*, 2014.

10 John McCarthy and Patrick Hayes. *Some philosophical problems from the standpoint of artificial intelligence*. Stanford University USA, 1968.

11 Federico Peinado, Marc Cavazza, and David Pizzi. Revisiting character-based affective storytelling under a narrative bdi framework. In *Interactive Storytelling*, pages 83–88. Springer, 2008.

12 Mark O Riedl and R Michael Young. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Generation Computing*, 24(3):303–323, 2006.

13 Mark O Riedl and Robert Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39(1):217–268, 2010.

14 Ivo Swartjes and Mariët Theune. A fabula model for emergent narrative. In *Technologies for Interactive Digital Storytelling and Entertainment*, pages 49–60. Springer, 2006.

15 John Ronald Reuel Tolkien. *The Lord of the Rings: One Volume*. Houghton Mifflin Harcourt, 2012.

16 Tom Trabasso, Paul Van den Broek, and So Young Suh. Logical necessity and transitivity of causal relations in stories. *Discourse Processes*, 12(1):1–25, 1989.

17 Mischa M Tuffield, Dave E Millard, and Nigel R Shadbolt. Ontological approaches to modelling narrative. In *Proc. of the 2nd AKT DTA Symposium*, 2006.

18 Johan Van Benthem. Modal foundations for predicate logic. *Logic Journal of IGPL*, 5(2):259–286, 1997.

Schemas for Narrative Generation Mined from Existing Descriptions of Plot*

Pablo Gervás[1], Carlos León[2], and Gonzalo Méndez[2]

1 Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid
Ciudad Universitaria, 28040 Madrid, Spain
pgervas@ucm.es

2 Facultad de Informática, Universidad Complutense de Madrid
Ciudad Universitaria, 28040 Madrid, Spain
{cleon,gmendez}@ucm@ucm.es

Abstract

Computational generation of literary artifacts very often resorts to template-like schemas that can be instantiated into complex structures. With this view in mind, the present paper reviews a number of existing attempts to provide an elementary set of patterns for basic plots. An attempt is made to formulate these descriptions of possible plots in terms of character functions, an abstraction of plot-bearing elements of a story originally formulated by Vladimir Propp. These character functions act as the building blocks of the Propper system, an existing framework for computational story generation. The paper explores the set of extensions required to the original set of character functions to allow for a basic representation of the analysed schemata, and a solution for automatic generation of stories based on this formulation of the narrative schemas. This solution uncovers important insights on the relative expressive power of the representation of narrative in terms of character functions, and their impact on the generative potential of the framework is discussed.

1998 ACM Subject Classification F.4.1 Knowledge Representation Formalisms and Methods

Keywords and phrases Narrative generation, conceptual representation of narrative, character functions, plot, narrative schemas

Digital Object Identifier 10.4230/OASIcs.CMN.2015.54

1 Introduction

Computational generation of literary artifacts very often resorts to template-like schemas that can be instantiated into complex structures. This approach has been addressed in the story generation field as a number of computational systems following a grammar-based design [9, 6, 5].

With this view in mind, the present paper reviews a number of existing attempts to provide an elementary set of patterns for basic plots. None of these attempts have been accepted as generally valid. To a large extent, they rely on oversimplification – reducing plot to a very abstract outline that conforms to a great number of story but characterises none of them –, or they focus on particular aspects of a given story – to the detriment of others – so it can be reduced to a schema that matches a larger number of stories. Such characteristics may play against the usefulness of any particular one of them as single framework for the description or classification of stories. However, considered as a whole, they can be understood

* This work was partially supported by FP7 WHIM project Grant Agreement 611560.

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 54–71
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 The Seven Basic Plots as described by Booker.

Overcoming the Monster	hero sets out to confront a monster and eventually defeats it
Rags to Riches	hero starts from humble beginnings and eventually achieves happiness
The Quest	hero sets out to fulfill a quest
Voyage and Return	hero sets out on a journey and returns having matured in the process
Comedy	initial confusion involving love relationships is eventually resolved happily
Tragedy	traces the fall from grace of a particular character to a tragic ending
Rebirth	main character almost falls from grace but repents at the last minute

as a basic abstract vocabulary to describe different plots. In the context of automated story generation, such a vocabulary would be very useful in at least two different senses:

- it may provide an agreed vocabulary for describing what type of story is desired, e.g. "a vengeance story" or "a quest story"
- it may provide a basic skeleton that the desired story should satisfy, regardless of any additional complexity that may be introduced to enrich it

In order to address needs of this kind, the present paper attempts to formulate these descriptions of possible plots in terms of schemas that may be used to drive the Propper system, an existing framework for computational story generation. The paper also explores the set of extensions required to the original set of character functions to allow for a basic representation of the analysed schemata. This is intended as a proof of concept to test the initial hypothesis of the usefulness of such schemas in the context of story generation. The Propper system [3, 4] is a computational implementation of the procedure for generating stories described by Vladimir Propp [8] as a possible use of his classic formalization of the morphology of the folk tale.

Once the various descriptions for plot are available as schemas that can be used to drive the Propper system, the impact of using them instead of - or as well as - the original canonical sequence for folk tales is discussed in terms of whether it expands the generative potential of the Propper system.

2 Review of Previous Work

This section reviews some of the existing proposals for the schematisation of possible story plots, the Proppian morphology of a folk tale, and the Propper system for story generation. Later sections bring these ingredients together to propose a computational model of narrative that can consider input in terms of the reviewed plot schemas and produces matching stories.

2.1 Some Existing Descriptions of Schemas for Plot

Christopher Booker [2] proposes that there are seven basic plots such that all possible stories can be seen as instantiations of these. The seven plot in question are described briefly in Table 1. These descriptions attempt to capture the basic outline for purposes of reference, more detailed descriptions follow below.

Table 2 20 Master Plots as presented by Tobias.

Quest	hero sets out to fulfill a quest
Adventure	much like a Quest but with less focus on a particular goal and more action
Pursuit	hero is pursued and eventually manages to escape
Rescue	hero rescues a victim imprisoned by a villain
Escape	like Rescue but the protagonist is the victim and eventually escapes by his own means
Revenge	protagonist sets out to avenge a villainy
The Riddle	involves solving a riddle (reader should try to solve it before the protagonist)
Rivalry	a protagonist and an antagonist of balanced power clash, protagonist wins
Underdog	as in Rivalry but protagonist is at disadvantage and wins through tenacity
Temptation	maps the fight of protagonist against temptation, from initial fall to eventual success
Metamorphosis	protagonist suffers a curse that transforms him into a beast, but love releases him eventually
Transformation	faced with a crisis, protagonist suffers transformation with important effects (usually at a price)
Maturation	tracks immature character through challenging incidents to maturity (usually achieved at a price)
Love	maps the progress of a love relation from initial obstacles to final fulfillment (if test passed)
Forbidden Love	as in Love but around an unconventional love relation (usually adultery) which ends badly
Sacrifice	tracks transformation of main character from low to high moral state, leading to a final sacrifice
Discovery	protagonist discovers himself
Wretched Excess	traces psychological decline of a character based on a character flaw
Ascension	protagonist faces a moral dilemma and undergoes ups and down till he reaches success
Descension	as in Ascension but followed to final disaster

An important point to note is that these plots are not mutually exclusive. Any given narrative may combine several of them into its overall structure, with some of these subplots possibly focusing on different characters.

Tobias [10] proposes the existence of 20 master plots. His book is more oriented towards instruction on how to build instances of these plots. A relevant insight presented here is that plots can be divided into plots of the body – involving mainly action – and plots of the mind – involving psychological development of the characters. Brief descriptions of these 20 master plots are provided for reference in Table 2.

The 20 plots by Tobias are even more difficult to keep separate from one another in practical terms. In terms of actual events in the narrative, quests or adventures are very likely to include elements of pursuit, rescue, escape, rivalry, revenge, temptation, sacrifice, or some character being an underdog at some stage. In terms of character development, they may also include transformation, maturation, or discovery. Much the same may be said about love stories. Our understanding it that a plot is considered to satisfy one of these labels only if the label is applicable to the main structure of the plot.

Georges Polti [7] proposed 36 dramatic situations, following Gozzi's assertion that there can only be thirty six tragic situations. These situations are briefly described for reference in Table 3, although Polti divides each of them into a series of classes and sub-classes that are further described or exemplified in the referenced book.

These 36 situations can be combined in the same story, since they must be understood as an outcome of previous events in the story, when the intervening characters come together and the main character in the situation must face a decision to be made, a change to be suffered or an obstacle to be overcome.

2.2 Proppian Morphology of a Story

At the start of the 20th century, Vladimir Propp [8] identified a set of regularities in a subset of the corpus of Russian folk tales collected by Afanasiev [1]. These regularities he formulated in terms of *character functions*, understood as acts of the character, defined from the point of view of their significance for the course of the action. Character functions are so named because, in Propp's understanding, they represent a certain contribution to the development of the narrative by a given character. According to Propp, for the given set of tales, the number of such functions was limited, the sequence of functions was always identical, and all these fairy tales could be considered instances of a single structure.

The set of character functions includes a number of elements that account for a journey, a number of elements that detail the involvement of the villain – including the villainy itself, some possible elaborations on the struggle between hero and villain, and a resolution –, a number of elements that describe the dispatching of the hero, a number of elements that describe the acquisition of a magical agent by the hero, and a number of elements concerned with the progressive unveiling of the hero's role in opposition to a false hero.

It is less well known that Propp provides in his book a very clear description of how his morphology could be used for story generation.

2.3 The Propper System

The Propper system developed by Gervás [3] constitutes a computational implementation of a story generator initially based on Propp's description of how his morphology might be used to generate stories.

It relies on the following specific representations for the concepts involved:

- a *character function*, a label for a particular type of acts involving certain named roles for the characters in the story, defined from the point of view of their significance for the course of the action
- a sequence of character functions chosen as backbone for a given story
- possible instantiations of a character function in terms of specific *story actions*, involving a number of *predicates* describing events with the use of *variables* that represent the set of characters involved in the action

Based on these representations the Propper system defines a procedure that first chooses a sequence of character functions to act as abstract narrative structure to drive the process, and then progressively selects instantiations of these character functions in terms of story actions to produce a conceptual representation – in terms of an ordered sequence of predicates – of a valid story.

Table 3 The 36 dramatic situations as described by Polti.

Supplication	power in authority must choose between a persecutor and a suppliant
Deliverance	protector comes to the rescue of the distressed
Crime Pursued by Vengeance	avenger executes a vengeance on a criminal
Vengeance taken for kindred upon kindred	avenger and the criminal are kin
Pursuit	hero is pursued by an abstract peril or punishment
Disaster	a power is defeated by an enemy or catastrophe
Falling Prey to Cruelty of Misfortune	hero suffers a cruel master or misfortune
Revolt	hero is a conspirator that intrigues against a tyrant
Daring Enterprise	hero attempts to recover an object or person from an adversary
Abduction	hero rescues an abducted victim from its abductor
The Enigma	a combat of the intelligence to find a person or object
Obtaining	aim to be achieved through eloquence and diplomacy
Enmity of Kinsmen	kinsmen transform love into (usually) mutual hatred
Rivalry of Kinsmen	a desired person causes a kinsman to hate another
Murderous Adultery	a betrayed husband or wife kills one or both adulterers
Madness	a madman slays, injures or brings disgrace onto a victim
Fatal Imprudence	imprudence or curiosity as the cause of a loss
Involuntary Crimes of Love	character unknowingly commits adultery or incest
Slaying of a Kinsman Unrecognized	unrecognized victim is slain by a kinsman
Self-Sacrifice for an Ideal	hero sacrifices life, love or well-being to a cause
Self-Sacrifice for Kindred	hero makes sacrifices for happiness of a relative
All Sacrificed for Passion	character makes sacrifices for a vice or passion
Necessity of Sacrificing Loved Ones	hero sacrifices a loved one for a necessity or vow
Rivalry of Superior and Inferior	two masculine or feminine rivals with different rank
Adultery	a deceived husband or wife
Crimes of Love	a lover and beloved incur in questionable acts
Discovery of the Dishonor of a Loved One	a character discovers the shame of a loved one
Obstacles to Love	marriage prevented by social norms
An Enemy Loved	one of two lovers is hated by kinsmen of the other
Ambition	character tries to obtain a good guarded by an adversary
Conflict with a God	a mortal struggles with a deity
Mistaken Jealousy	a character is jealous of another
Erroneous Judgement	any kind of mistaken judgement
Remorse	a culprit suffers remorse for a crime or love fault
Recovery of a Lost One	a hero struggles to find a lost, loved one
Loss of Loved Ones	a character witnesses the death of a loved one

Table 4 Set of character functions employed as canonical sequence.

test by donor
hero reaction
acquisition magical agent
villainy / lack
hero dispatched
begin counteraction
acquisition magical agent
departure
test by donor
hero reaction
acquisition magical agent
transfer
trigger resolved
unrecognised arrival
unfounded claims
struggle
difficult task
branding
victory
task resolved
trigger resolved
return
hero pursued
rescue from pursuit
unrecognised arrival
unfounded claims
false hero exposed
transfiguration
branding
villain punished
hero marries

To fulfill Propp's description of the morphology of a folk tale, the sequence of character functions that acts as backbone for a story has to be a subset of the character functions listed by Propp, appearing in a relative order that conforms with a given canonical sequence. The actual set of character functions employed as canonical sequence is given in Table 4. Character functions are presented in two columns by their abbreviated name. A key point in the canonical sequence is the **villainy / lack** pair of character functions written in bold. These differ from all the others in that only one of them is ever included in any single story, and all stories must contain either one or the other.

From a given sequence of character functions, the system defines a *fabula*, a sequence of states that contain a chain of story actions – which are instances of those character functions. A story action involves a set of preconditions – predicates that must be present in the context for continuity to exist –, and a set of postconditions – predicates that will be used to extend the context if the action is added to it. Each story action is linked to its context of occurrence by having its preconditions satisfied by the preceding state. The initial state by default incorporates all predicates of the first action, and each valid action added to the fabula generates a new state that incorporates all predicates of the previous state, plus the predicates of the new action. To evaluate whether the preconditions of a story action are satisfied by the context, they are unified with the set of predicates that hold in that state.

The revised version described in [4] describes extensions to the original constructive procedure that take into account the possibility of dependencies between character functions – such as for instance, a kidnapping having to be resolved by the release of the victim – and the need for the last character function in the sequence for a story to be a valid ending for it.

3 Describing Existing Schemas for Plots in Terms of Proppian Character Functions

We want to attempt to unify the material reviewed in Section 2 into a single representation that is compatible with the existing framework of the Propper system. As the Propper system is driven by Proppian character functions, we will consider whether the schemas arising from the approaches reviewed can be described as sequences of character functions as described by Propp, and what extensions might be required for a better fit.

3.1 Establishing a Common Vocabulary from the Set of Taxonomies

The different sets of plots reviewed in Section 2.1 show a certain overlap in some cases (both Booker and Tobias include a plot based on a quest, for instance). Where they differ, it would be ideal to establish some way in which the elements in one set might be related to elements in the other, either as more specialised or more abstract versions.

When trying to cross-relate these various taxonomies with one another, it becomes apparent that they are formulated at different levels of abstraction, and focused on different aspects of the plot. This makes it difficult to find a clear correlation between them. However, for the purposes of our paper – which aims at making it possible to rely on these descriptions to specify desired stories and/or drive the process of their construction – it becomes important to be able to understand how elements from these descriptions might combine or interact.

In that sense, a number of patterns can be identified. Tobias' and Booker's plots can be related as follows:

- Tobias' plots of Temptation, Metamorphosis, Transformation, Maturation and Discovery could fit Booker's description of Rebirth plots.
- Tobias' plots of Pursuit, Rescue, Escape, Rivalry, Underdog, Revenge, Sacrifice might be employed to articulate what Booker describes as an Overcoming the Monster plot.
- Tobias' Love plot correlates nicely with Booker's Comedy plot.
- Tobias' plots of Wretched Excess, Descension, Forbidden Love, and, possibly, Sacrifice might fit Booker's Tragedy plot.
- Tobias plot of Ascension fits Booker's Rags to Riches plot.
- Tobias' plots of Transformation, Maturation and Discovery could apply as descriptions of character development implicit in Booker's description of Quest, Voyage and Return, Rags to Riches and Rebirth plots.

Polti's dramatic situations are not presented as candidates for complete plots, but rather as situations with dramatic potential that may arise within a given plot. In this sense, they are easier to place with respect to the other two proposals considered in this paper. In a sense, they constitute a finer grained vocabulary for describing plot elements that may occur in larger plot structures. For this reason, some of them show a surprising match with those plots of Tobias' that we have described as elements sometimes used as ingredients being expanded into full independent plots, such as Pursuit – which appears in both Tobias' and Polti's lists –, or Deliverance in Polti closely matching Rescue in Tobias.

For this set of situations, the task to be considered becomes more to identify where in the more elaborate structures these situations appear.

3.1.1 Paraphrasing Plot Options in Terms of Character Functions

Booker's set of seven plots can be easily paraphrased in terms of Proppian character functions. One such paraphrase of them is given in Table 5. There are some differences. Where Propp considers a fixed sequence of character functions from which a selection can be picked out, Booker's descriptions differ in at least two ways. First, they sometimes allow for more than one possible relative ordering between some of the elements included. In the table, this has been represented by placing between brackets those elements that may occur in interchangeable order or that are optional. Second, Booker's descriptions include a certain possibility of some subsequences reoccurring repeatedly over the same plot. In the table,

Table 5 Paraphrases of Booker's 7 basic plots in terms of Proppian character functions.

Overcoming the Monster	(villainy*, MONSTERS*), struggle, victory, villain punished, hero marries
Rags to Riches	lack, departure, transfiguration, hero marries
The Quest	(hero dispatched, difficult task), departure, (MONSTERS*, HELPER*), task resolved
Voyage and Return	departure, ((difficult task, task resolved), (MONSTERS*, HELPER*)), return
Comedy	lack, (transfiguration, unrecognised arrival), (difficult task, task resolved)*, (hero recognised), transfiguration, hero marries
Tragedy	(villainy*, MONSTERS*), struggle, victory, villain punished
Rebirth	(villainy*, MONSTERS*), *repentance*, *repentance rewarded*

such subsequences have been replaced with labels in capital letters that have been defined separately. It may pay to abstract them into higher order labels that can appear within more structured sequences. They correspond to:

- MONSTERS: struggle, hero pursued, (victory, rescue from pursuit)
- TESTERS: test by donor, hero reaction, acquisition magical agent

Where certain character functions (or labels for subsequences) can occur more than once according to Booker, these have been marked with an asterisk *. The case of Tragedy and Rebirth is strikingly different. Both can indeed be phrased in terms of Proppian character functions as shown in the table. However, this requires a slight revision of the Proppian concept of character function. Proppian character functions assume a fixed set of roles, namely a hero, a villain and some auxiliary characters such as dispatcher, a donor, a helper... But in Proppian functions, the protagonist of the story is assumed to be always the hero. In the case of Booker's Tragedy and Rebirth, the paraphrase works only if the protagonist is considered to be the villain. This implies that the Tragedy plot would correspond to an instance of the Overcoming the Monster plot but told from the point of view of the villain. It is important to note that the occurrence of the victory character function now implies that the protagonist is defeated, which is contrary to Propp's original interpretation. The Rebirth plot requires a more elaborate reworking to be phrased in terms of Proppian functions, because it involves a particular turn in the story that was not originally contemplated by Propp. This is the point in the narrative where the villain sees the light, repents, and redeems himself. New character functions would need to be introduced to cover this process, as it plays a fundamental role in such stories that would definitely need capturing. We refer to these character functions as *repentance* and *repentance rewarded*, and we include them as such in the table.

The Comedy plot requires a special analysis. It may be phrased in terms of Proppian functions, in as much as it starts from an initial lack – though specifically related to love, lack of a love partner, lack of attention from the chosen partner, or lack of permission to marry the chosen partner –, it involves solving a difficult task – related to the corresponding lack –, and it ends with the hero marrying. However, the description of this plot provided by Booker addresses the corresponding story at a level of detail that cannot be covered appropriately with Proppian functions, at least in the sense that these had been defined within the Propper system. To deal with this case, we would need a system with the following features:

Table 6 Paraphrases of the Elementary Plots of Tobias' in terms of Proppian character functions.

Pursuit	hero pursued, rescue from pursuit
Rescue	villainy, trigger resolved
Escape	villainy, trigger resolved [protagonist is victim, not hero!]
Revenge	villainy, villain punished
The Riddle	difficult task, task resolved
Rivalry	struggle, victory
Underdog	struggle, victory [protagonist at disadvantage]

- the ability to explicitly represent the gender of characters[1], as the core of the plot revolves around love relations between characters
- the ability to represent shifts in affinity between characters and to have these shifts arising from and triggering events in the narrative
- the ability to consider a number of interwoven subplots focused on different characters

Such features are beyond the scope of the present paper but they will be considered for future work. Nevertheless, a basic sketch of the Comedy plot in terms of Proppian functions has been provided for completeness.

According to Booker's description of his plots, the paraphrases given in Table 5 constitute a sketch of the main events that characterise each of the plots. The fleshing out of these plots into specific actual stories may involve combining more than one plot, in which case the corresponding sequences of character functions may intermingle as different narrative threads. When such task is attempted computationally, some means must be provided for keeping track of which characters play which roles in which of these threads, and whether any given character can play different roles in different threads. This is beyond the scope of the present paper and it is currently left for future work.

As discussed in Section 3.1, the elements described by Tobias amongst his 20 master plots operate at a slightly different level of abstraction from those used by Booker. In a certain sense, they correspond to focusing the plot of a complete story on particular types of situation that were occurring as parts of the plots considered previously. The correspondences already established between Booker's and Tobias' plots introduce a change in the overall task definition. Given that many of the plot descriptions given by Tobias can be seen as specific instances of Booker's plots, it is less useful to paraphrase them in terms of Proppian functions – the paraphrase already given for the corresponding Booker plot might be used in each case – and it becomes more interesting to consider how the different instantiations that Tobias provides might be differentiated from one another in terms of a Proppian description (or what extensions of the Proppian implementation might be required to consider these plots).

Tobias' plots of Pursuit, Rescue, Escape, Rivalry, Underdog, Revenge, Sacrifice can be represented as more specific plots that focus on parts of the sequences of character functions used to describe Booker's plots. A tentative paraphrasing for them is presented in Table 6.

The Quest and Adventure plots can be seen as similar to Booker's The Quest and Voyage and Return. Tobias' Love plot has been linked to Booker's Comedy plot, and so it is subject

[1] Although in current times it might have been more politically correct to phrase this in terms of sexual preferences, we have opted in this desiderata for a more classical approach to character pairings in terms of gender. This might be revised in future work to allow for more generic and politically correct story telling capabilities.

to the same considerations described earlier for that one. The Ascension plot can be mapped to the Rags to Riches plot.

The remaining plots described by Tobias can be grouped into a set of instantiations of the two Booker plots already described that presented significant differences with the Proppian schema: Tragedy and Rebirth.

Forbidden Love is related to Comedy/Love plots in that its main ingredient is a love relationship, and it differs from them in two ways: the love relation in question is one against convention, and it ends badly. As before, this may be implemented using the same set of characters and actions as for comedy, but making the protagonists a pair of characters that do not get paired off in the end. This is similar to the opposition between Overcoming the Monster and Tragedy. In a sense, one could say that Tobias is enriching the set of plots by considering a plot based on love but which can end badly, whereas Booker only considers plots on love that end well.

In a similar opposition, the Descension and Wretched Excess plots could be seen as dark counterparts to the Rags to Riches/Ascension type of plot. These may be paraphrased in terms of Proppian functions by inverting the order in which the functions in the sequence for Rags to Riches occur. However, better results might be obtained if specific character functions are defined to represent: an initial positive situation for the character – corresponding to a positive version of lack –, a character function to discover events in which the fortune of the protagonist suffers, and a final negative situation. This suggests that a reworking of the set of character functions might benefit from a little generalization, so that both positive and negative situations can be described, and events that cause transitions in both positive and negative directions can be represented. Then the opposing pairs of plots may all be represented based on these. The original set of character functions defined by Propp covers only part of this spectrum – it includes no character function for a positive initial situation – and relies on very specific solutions for some particular areas – it links very tightly the final positive situation of the hero with either marriage or coronation, for instance. An effort to broaden this set of character functions would greatly improve the range of possible stories that can be generated. As this requires a heavy effort of knowledge engineering of system resources it is postponed for future work.

Differences between Descension and Wretched Excess can be identified in terms of one being more concerned with material situation of the protagonist, and the other with his/her psychological decline. In marking this difference, Tobias shows a concern with an aspect of plots that had not been considered by either Propp or Booker: the difference between physical and psychological characterization.

The set of plots proposed by Tobias shows an increase in number partly because it distinguishes a number of plots that are based on psychological development of their protagonists – what he describes as plots of the mind – beyond those considered by Propp – which centre almost exclusively on what Tobias calls plots of the body. These plots of the mind are the Temptation, Transformation, Maturation and Discovery plots. The Metamorphosis plot combines such a psychological ingredient with a physical change. In terms of Booker's classification, most of these qualify as Rebirth plots, as they involve a change of the protagonist during the development of the plot. In a certain sense, the Sacrifice plot also includes a similar turning point related to psychological issues, though in this case the change also translates into a physical sacrifice. The differences between the various plots arise from these slight differences in the relative importance of the material and the psychological aspects, or in the specific type of change that the protagonist is subjected to – as described reasonably well by the names of these plots.

Again, the representation of the psychological evolution of characters is beyond the current capabilities of the Propper system, and discussion of an appropriate extension beyond the scope of the present paper, but it will be considered as future work.

With respect to Polti's dramatic situations, these are not so much patterns for complete plots but rather building blocks that may be employed in the construction of plots. In this sense, they are closer to being descriptions of actions of the characters that are significant for the course of the action, which is what Propp's character functions are intended to be. For this reason, when establishing a correspondence that might lead to a common vocabulary for plot descriptions, it would be more useful to consider Polti's dramatic situations as alternative abstractions, closely related to Proppian character functions. A possible alignment between Polti's dramatic situations and Propp's character functions (or groups thereof) is shown in Table 7. The material is presented according to the following criteria. For each line of the table, the first column indicates a character function or a group of character functions that might be considered to correlate in some way with the dramatic situations listed in the second column. The third column is used to indicate specific characteristics that the instantiations of the character functions given in the first column would need to satisfy to properly represent the dramatic situation given in the second column. The bottom half of the table shows dramatic situations that have no direct match to Proppian character functions. For these, it may be worth considering the introduction of specific character functions.

3.2 Extending the Propper System for Schema-Driven Generation

Once a common vocabulary has been agreed that includes elements from the various taxonomies, the Propper system has been extended to take advantage of it.

This implies two basic extensions beyond the previous versions of the system:

- it must accept input in the form of elements from this vocabulary to drive the story that is to be constructed
- it must be capable of producing stories that match the corresponding description

The first extension has been achieved by means of a preprocessing module that, given the name of a given narrative schema, builds a sequence of character functions based on resources along the lines of the tables presented in Section 3.1.1. To build a proof of concept, the complexities of repetition and alternative ordering have not been considered and the initial version focuses on simple instantiations of the more generic sequences. These sequences can now be used as input to the stage of fabula generation of the Propper system, which searches for appropriate instantiations of these character functions in terms of story actions that link into a coherent whole that can be recognisable as a story.

The second extension has proven to be more difficult, but it has also uncovered a number of important insights on the advantages and disadvantages of Propp's framework as a computational model of narrative. Additionally, this effort has prompted a number of improvements that have allowed the system to go beyond Propp's original formulation.

The first insight relates to the fact that most of the sequences required to implement the set of narrative schemas reviewed were already included in the canonical sequence proposed by Propp. This must be considered an important merit of Propp's framework as it implies that the method for story generation outlined by Propp – in terms of selecting character functions from his canonical sequence and instantiating them – would in theory be capable of producing instances of most of the narrative schemas reviewed. The difficulty would lie in how to inform the choices at each point. This is part of the problem that the rest of this section attempts to address.

Table 7 Alligment of Polti's 36 Dramatic Situations with Proppian character functions.

lack	Ambition Recovery of a Lost One Loss of Loved Ones	
lack villainy	Disaster Falling Prey to Cruelty of Misfortune Madness Fatal Imprudence Involuntary Crimes of Love Slaying of a Kinsman Unrecognized Adultery Crimes of Love Discovery of the Dishonor of a Loved One	 (love) (love)
trigger resolved rescue from pursuit	Deliverance	
victory villain punished trigger resolved	Crime Pursued by Vengeance Vengeance taken for kindred upon kindred	
hero pursued	Pursuit	
struggle	Enmity of Kinsmen Rivalry of Kinsmen Rivalry of Superior and Inferior	(psychological)
trigger resolved	Abduction Murderous Adultery	
test by donor hero reaction acquisition / difficult task task resolved	Daring Enterprise The Enigma (temptation or a riddle) Obtaining	
	Self-Sacrificing for an Ideal Self-Sacrifice for Kindred All Sacrificed for Passion Necessity of Sacrificing Loved Ones	(sacrifice) (sacrifice) (sacrifice) (sacrifice)
	Obstacles to Love An Enemy Loved	(love) (love)
	Mistaken Jealousy Erroneous Judgement Remorse	(psychological) (psychological) (psychological)
	Supplication Revolt Conflict with a God	

The second insight concerns the fact that the set of story actions developed to cover the Proppian character functions includes a broad range of possible story actions to instantiate each character function. However, in many cases the specific instances of character function occurring in the context of one of these more specific narrative schemas need to be restricted to a subset of the complete range of possible story actions. For instance, when the character function for *lack* occurs at the beginning of a Rags to Riches schema it works better if instantiated with story actions concerned with hardship or poverty rather than desire for wondrous magical objects, whereas both occur in the context of Proppian tales. When the same character function occurs at the beginning of a Comedy plot, it only works if instantiated with story actions concerned with lack of a love partner, or lack of permission to marry. To address this issue, the module of the Propper system concerned with retrieving possible story actions to instantiate a given character function has been refined to take into account what particular narrative schema is being considered in each case. The knowledge of which story actions are suitable to instantiate which character functions under particular narrative schemas has been encoded explicitly in resources local to these modules. A similar mechanism may be applied to address the more detailed specific instantiation of character functions required to generate instances of Tobias's plots and/or Polti's dramatic situations, as described above.

A third important insight arose from the observation that, whereas the Proppian morphology takes for granted that the protagonist of the stories is always the hero, some of the set of narrative schemas considered focused on the villain as protagonist. Namely, Booker's schemas for Tragedy and Rebirth, and those of Tobias's plots that in the analysis in Section 3.1 have been associated to these two. This presents no problem to our endeavour in as much as the conceptual representation of a story as currently produced by the Propper system is agnostic as to who is the protagonist. This will become apparent in the examples presented later in the paper. This issue of who the protagonist is would have to be taken into account in future work, once the problem of rendering these conceptual representations of stories as text is addressed.

A fourth insight focused on the fact that to obtain sequences of character functions that matched as closely as possible the descriptions of the narrative schemas, certain character functions (or subsequences thereof) might need to occur more than once. This presented problems because not all instances of the available story actions allowed this. For instance, some of the story actions for the *victory* character function allowed the villain to survive the encounter – thereby being available for a second struggle later in the story –, whereas others ended more radically with his demise. This restriction was particularly important to distinguish between the two types of schema where the villain acts as protagonist of the story: instances of the Rebirth narrative schema require the villain to repent at some point in the story and undergo a radical change for good, whereas instances of Tragedy may well end in his utter destruction. From a computational point of view, it required a solution capable of discerning which particular story actions could be used to instantiate a character function at what points of the story. The process for selecting story actions was refined further to take into consideration the relative position of each character function within the narrative schema being considered.

The observed possibility of repeating and combining certain subsequences of character functions to make up more complex schemas led to a fifth insight concerning Propp's morphology. Although the canonical sequence of character functions as described by Propp includes a certain redundancy to allow character functions (or small subsequences of them) to occur at more than one point in the overall narrative arch, the morphology as formalised is too

Table 8 An example story for the Overcoming the Monster narrative schema.

0	character id810
0	torment_at_night id810 id811
0	victim id811
0	character id811
0	misbehaved id810
1	runs_away id811
1	pursues id810 id811
1	demands id810 id811
2	hides id316 id811
2	escapes id811
3	weight_contest id811 id810
3	confrontation id811 id810
4	heavier id811
5	punished id810
5	shot id810
6	marries id811
6	acceeds_to throne id811

rigid to capture appropriately the broad range of narrative schemas that have been reviewed. Propp's insistence that the character functions in his morphology need be considered in a specific order introduces a restriction that reduces the expressive power that it might otherwise have had. This is particularly relevant given that the set of narrative schemas reviewed is by definition a subset of all the possible ones. For this reason, we intend to address as future work alternative possible means of combining these sequences of character functions into complex narrative schemas.

3.3 Examples of Constructed Stories Matching Given Narrative Schemas

Although it would be impossible to include in this paper examples of stories to match all the various narrative schemas reviewed, an effort has been made to cover instances of at least the seven basic plots described by Booker. As the other narratives schemas or dramatic situations have been related back to these seven in the sections above, this should be seen as an indication of the potential of the approach.

The task of extending the knowledge resources of the system to cover the full set of schemas would be significant. The original knowledge engineering effort for the first version of the Propper system, as reported in [3], demonstrated this task to be an important bottleneck for the development of this type of system. As a proof of concept, a basic initial version of the desired approach has been implemented based on the existing resources in terms of related sets of character functions and story action resources. The two new character functions *repentance* and *repentance rewarded* and a small set of possible instantations of them as story actions have been added. The stories that result from this effort are reported below.

Table 8 presents an example of story corresponding to the Overcoming the Monster narrative schema. This particular story has the peculiarity that the system has picked the victim of the initial villainy as the hero of the story.

Table 9 presents an example of story corresponding to the Rags to Riches narrative

Table 9 An example story for the Rags to Riches narrative schema.

0	character id301
0	lack id301 money
1	sets_out id301
2	builds id301 palace
2	new_physical_appearance id301
3	marries id301

Table 10 An example story for the Comedy narrative schema.

0	character id298
0	lack id298 bride
1	puts_on id298 garment
1	deceiving_appearance id298
2	arrives id298 id719
2	location id719
2	disguised id298
2	unrecognised id298
3	sets id157 id298
3	character id157
3	involves difficult_task hiding
4	solve id298 difficult_task
4	before dead_line
5	recognised id298
6	puts_on id298 garment
6	new_physical_appearance id298
7	betrothed id298

schema. This story is indicative of how the simplest structure that conforms to one of these schemas may be insufficient to hold the reader's interest and fleshing out with additional narrative elements may be required.

Table 10 presents an example of story corresponding to the Comedy narrative schema. As indicated above, this is intended only as a baseline. Quality would improve significantly once the complexities outlined earlier as required for Comedy are addressed.

Table 11 presents an example of story corresponding to the Tragedy narrative schema. It is important to note that in this story the protagonist must be considered to be character id775, who plays the role of the villain.

Table 12 present an example of story corresponding to the Rebirth narrative schema. Again, the protagonist of this story is character id805.

The stories for narrative schemas corresponding to The Quest and Voyage and Return as described rely heavily on a combination of a number of incidents. As a result, they turned out to be overlong to be reported within the size limitations of the paper, but the system has been extended to be able to produce them. They also suffer from the rigid sequencing of the various elements involved (struggles with villains, chases, task to solve, encounters with magical helpers). The more flexible solution for the relative ordering of these elements that is being considered as future work would result in better stories.

Table 11 An example story for the Tragedy narrative schema.

0	character id775
0	substitute id775 id776 id777
0	victim id776
0	character id776
0	bad id777
0	misbehaved id775
1	runs_away id776
1	pursues id775 id776
1	demands id775 id776
2	throws id776 id310
2	turns_into id310 id312
2	obstacle id312
2	escapes id776
3	weight_contest id776 id775
3	confrontation id776 id775
4	heavier id776
5	punished id775
5	shot id775

Table 12 An example story for the Rebirth narrative schema.

0	character id805
0	try_to_eat id805 id806
0	victim id806
0	character id806
0	misbehaved id805
1	runs_away id806
1	pursues id805 id806
1	demands id805 id806
2	turns_into id806 id314
2	unrecognisable id314
2	escapes id806
3	play id806 id805 cards
3	confrontation id806 id805
4	wins id806
5	repents id805
6	acceeds_to throne id805

4 Discussion

The extensions that have been required to enable the representation of existing plot schemas as paraphrases in terms of Proppian character functions arose from one of two possible situations:

- the plots in question violated one of Propp's basic premises (which basically involve the protagonist being the hero and the tale having a happy ending)
- the set of character functions did not allow a direct representation of some complication in the plot

The first situation has been easily resolved by allowing the story generation to consider stories that violate Propp's premises. Once the roles in the story have been decoupled from the choice of protagonist, the existing set of character functions allows representation of different stories simply by shifting the protagonism to characters that do not succeed in the end. These have always existed as antagonists, and they can now become protagonists of tragic stories.

The second situation has consequences at two different levels. First, the Proppian set of character functions did not contemplate complications like fluctuating love relations or psychological development of characters. The multiplication of the number of possible schemas for plot arise from the consideration of instances of particular subsequences that present specific characteristics related to these features not contemplated by Propp. Some of these complications required a significant overhaul of the expressive power of the underlying computational system and can only be considered as further work.

Yet other complications would require only a dual process of generalization/instantiation of the character functions in the existing set to cover the missing features. Propp's set of character functions was developed for a very specific set of folk tales and it was not intended to be generalizaded beyond it. The concept of character function itself, in contrast, was defined as a generic tool for the analysis of narrative.

An extended set of character functions, satisfying Propp's requirements on the definition of a character function but covering the range of basic complications outlined in the present paper would be significant contribution to the field of narrative generation. The set of character functions developed by Propp has been tested repeatedly as a possible resource on which to base generic story telling system and has been found wanting [11]. The proposed extension might help to reduce the shortcomings perceived and increase the expressive potential of system based on a character function representation.

A further extension being contemplated as future work concerns the need for a flexible mechanism for combining meaningful sequences of character functions into larger narrative units, which would allow the system to capture more faithfully a larger set of the reviewed narrative schemas. A grammar-based solution such as the one outlined in [3] is being considered as a possible solution.

5 Conclusions

A number of existing descriptions of plot has been reviewed, and the resulting analises have been correlated to distill a basic vocabulary of narrative schemas. These narrative schemas have been paraphrased in terms of sequences of character functions as described in Propp's morphology. This has allowed the extension of an existing story generation system to generate output stories corresponding to the desired narrative schemas.

Important insights on the expressive power of Propp's morphology, and some discussion of its limitations as a generic story generation framework have been outlined. Limitations of Propp's morphology have been identified at three different levels. First, the sequencing and ordering of plot bearing elements/character functions as determined by Propp's formalism is too rigid to capture the flexibility of plots beyond Russian folk tales. Second, the set of abstractions for plot bearing elements/character functions would need to be extend, both with new elements and with additional annotations to existing ones, for instance regarding issues like gender of the characters, whether they survive the event, or whether the outcome is positive or negative for them. Third, an additional level of information concerning affinities between characters and/or psychological characteristics of the characters may need to be considered for dealing with Comedy plots as described by Booker or plots of the mind as described by Tobias.

The work reported in the paper is preliminary and ongoing, and several avenues of future work have been described. Some of these hold significant potential for improving both the quality of the resulting stories and the value of the proposed solution as a computational model of narrative.

Acknowledgements. This paper has been partially supported by the project WHIM 611560 funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

References

1 Alexander Nikolayevich Afanasyev. *Narodnye russkie skazki A. N. Afanaseva [Folk Russian tales of A. N. Afanasev]*, volume 1-3. Moscow, Nauka Publ, Moscow, 1855.

2 C. Booker. *The Seven Basic Plots: Why We Tell Stories.* The Seven Basic Plots: Why We Tell Stories. Continuum, 2004.

3 Pablo Gervás. Propp's morphology of the folk tale as a grammar for generation. In *Workshop on Computational Models of Narrative, a satellite workshop of CogSci 2013: The 35th meeting of the Cognitive Science Society*, Universität Hamburg Hamburg, Germany, 08/2013 2013. Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany, Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany.

4 Pablo Gervás. Reviewing Propp's story generation procedure in the light of computational creativity. In *AISB Symposium on Computational Creativity, AISB-2014, April 1-4 2014*, Goldsmiths, London, UK, 04/2014 2014.

5 S. Imabuchi and T. Ogata. A story generation system based on Propp combined with a conceptual dictionary. In *Natural Language Processing andKnowledge Engineering (NLP-KE), 2011 7th International Conference on*, pages 359–362, Nov 2011.

6 R. Raymond Lang. A declarative model for simple narratives. In *Proceedings of the AAAI Fall Symposium on Narrative Intelligence*, pages 134–141. AAAI Press, 1999.

7 G. Polti and L. Ray. *The Thirty-six Dramatic Situations.* Editor Company, 1916.

8 Vladimir Propp. *Morphology of the Folk Tale.* Akademija, Leningrad, 1928.

9 D. E. Rumelhart. Notes on a schema for stories. *Representation and Understanding: Studies in Cognitive Science*, pages 211–236, 1975.

10 R.B. Tobias. *20 Master Plots: And How to Build Them.* F+W Media, 2012.

11 Zach Tomaszewski and Kim Binsted. The limitations of a Propp-based approach to interactive drama. In *Intelligent Narrative Technologies, Papers from the 2007 AAAI Fall Symposium*, pages 167–173. AAAI Press, 2007.

Imaginative Recall with Story Intention Graphs

Sarah Harmon and Arnav Jhala

Department of Computer Science, University of California at Santa Cruz
Santa Cruz, USA
jhala@soe.ucsc.edu, smharmon@ucsc.edu

Abstract

Intelligent storytelling systems either formalize specific narrative structures proposed by narratologists (such as Propp and Bremond), or are founded on formal representations from artificial intelligence (such as plan structures from classical planning). This disparity in underlying knowledge representations leads to a lack of common evaluation metrics across story generation systems, particularly around the creativity aspect of generators. This paper takes Skald, a reconstruction of the Minstrel creative story generation system, and maps the representation to a formal narrative representation of Story Intention Graphs (SIG) proposed by Elson et al. This mapping facilitates the opportunity to expand the creative space of stories generated through imaginative recall in Minstrel while maintaining narrative complexity. We show that there is promise in using the SIG as an intermediate representation that is useful for evaluation of story generation systems.

1998 ACM Subject Classification I.2.7 Natural Language Processing: Discourse

Keywords and phrases Story generation, computational creativity, narrative, story intention graph

Digital Object Identifier 10.4230/OASIcs.CMN.2015.72

1 Introduction

Storytelling and creativity are key aspects of human cognition. While much work has been done on computational narrative generation, the focus of this research in recent years has been more toward generation of coherent sequences of events. Minstrel, one of the earliest story generators, utilized a case-based reasoning approach to incorporate a model of human creativity [17]. In this paper, we extend a contemporary rational reconstruction of Minstrel called Skald [16] by organizing and labeling story events. We then present a mapping between the underlying story representation in Skald to the Story Intention Graph (SIG) formalism proposed recently by [4], which is rooted in story understanding. This mapping and extensions to Skald allow us to identify areas of research that are unexplored both in terms of storytelling and creative systems.

Minstrel relies heavily on a library of cases, and employs a boredom mechanic which, although designed to generate more interesting results, quickly exhausts its library of reference stories. Considerable manual authoring is thus required as part of the original Minstrel system. There is also, notably, no reliable bridge towards a natural language generation system for a generic Minstrel-like program. As such, current attempts to expand the creative power of Minstrel produce graphs, rather than text which reads like a natural story [16]. Finally, it is difficult to compare storytelling systems like Minstrel with each other, because there is no definitive standard designed to assess the quality or scope of generated creative content. Here, we propose that a semantic representation system – the Story Intention Graph (SIG) model [4] – be used as a formalized standard of narrative meaning and comprehension.

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 72–81
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

With the adoption of this standard, generated narrative content, such as that composed by Minstrel, can be more easily analyzed, upgraded, and rewritten as natural text.

The SIG formalism provides several affordances that improve the richness of representation of stories beyond the parameterized case frames of situations. First, it is based on a rich model of internal states of agents involved in the narrative using a theory of mind approach. This approach maintains local coherence for characters while ensuring global coherence of the overall narrative. Second, it has a notion of a *plot unit* but at a richer level of semantic interconnections across plot units. Finally, the SIG representation provides a way to detect and reason analogies through metrics derived from the encodings. This is an important affordance, particularly for CBR-based generation systems.

The overall contributions of this work are two-fold. The primary contribution is the implementation of the SIG formalism in a case-based story generation system. The secondary contribution is the implementation of extensions to Minstrel's generation process in terms of event ordering and using a richer story representation to increase the expressive range of creative stories generated by the system.

2 Related Work

One of the first automated storytelling systems known was a murder mystery generator called Novel Writer [9]. The domain of generated stories for Novel Writer was very small: only one type of story was generated, and always involved a murderer, a motive, and someone who revealed the murderer. Further, the Novel Writer ruleset was highly constraining – allowing, for instance, only four possible motives for murder – and prevented the overall system from reaching a high level of creativity and expression.

Several years later, a system called TALE-SPIN [10] took a character-driven approach to story generation. In TALE-SPIN, multiple characters could develop plans to pursue individual-level goals. Additionally, characters had personalities and dynamic relationships with each other. Although revolutionary in terms of its character planning system, TALE-SPIN was criticized for not providing a model for the author's creative process and goals.

The AUTHOR program [3] was created for precisely this purpose. AUTHOR generated stories by simulating the intentions of a human author and striving to satisfy them. However, AUTHOR was designed with the underlying assumption that all generated narrative sequences must conform to a strict ruleset detailing story parameters and narrative structure. Within the AUTHOR system, then, there is not much freedom in terms of computational creativity.

The focus of modern systems is specifically on generation of plot structures (in plan-based approaches), drama management for sequencing predefined beat structures, or manipulating surface level discourse elements like language and visuals. The goal in these systems is either coherence of stories or management of player experience. While outputs of these generators do qualify as being creative, it is difficult to evaluate the systems in terms of creativity due to the variety of underlying representations and lack of an explicit model of creativity. Detailed review of modern storytelling systems is outside the scope of this paper as the primary focus is a discussion of creativity within a rational reconstruction of the classic story generation system.

3 Research Foundation

3.1 Minstrel, a Case-Based Reasoning Approach

Turner created the Minstrel [17] story generation system that takes a case-based reasoning approach to creative authoring of stories. Minstrel is a LISP program that simulates the

Table 1 A quantitative comparison between Minstrel Remixed and Skald. By using weighted TRAM searching and a modified boredom algorithm, Skald optimized TRAM results in terms of speed and retrieval quality.

Measure	Minstrel Remixed	Skald
TRAM search failure rate	19%	3.5%
Average number of TRAMs tried per search	58	16
Average number of TRAMs used when no direct match found	2.4	1.4

actions of a human author in order to produce stories. In particular, Minstrel models the human creative process by transforming memories of known events (case base) to formulate new scenarios via generalization and adaptation (referred to as *imaginative recall* in the original Minstrel description). Story elements are defined by schemas (case frames) and stored in a searchable database, and creating small changes in these schemas results in new stories.

To create new stories from prior examples, Minstrel relies on twenty-five heuristics called TRAMs ('Transform-Recall-Adapt Methods'). As an example, Minstrel contains a default TRAM called 'Standard-Problem-Solving' which simply looks for a pre-existing solution in memory. If no solution exists, the TRAM fails. The TRAM also fails if any found solutions have already been used, because such solutions are deemed 'boring' by the Minstrel system. Whenever a given TRAM fails, the problem must be transformed and Minstrel must look for a case that best matches the newly transformed problem.

3.2 Skald: Improving Minstrel's imaginative recall system

Skald[15] was developed to make the Minstrel system more robust and useful as a general-purpose story generator. While Minstrel applied TRAMs randomly, Skald employs a weighted TRAM searching algorithm which gives preferences to TRAMs that best match the original query. This technique reduces the search space, resulting in faster and higher quality generations (refer to Table 1). Skald also modifies Minstrel's boredom algorithm by only fractionally decrementing boredom signature values, enabling signatures to refresh over time and be reused in later stories. Although more 'interesting' stories are not forcibly produced as quickly as they would be in Minstrel, this technique traverses through the story library more slowly and makes more efficient use of the searchable domain. More stories can thus be produced with less manually-authored templates.

In Skald, groups of *symbols*, the most basic story elements, are grouped into *frames*. Frames may contain empty or unknown symbols (refer to Table 2). Groups of frames form an output story graph. Story characters have mental target objectives called *goals*, physical actions called *acts*, and *states*, which are results of action. Similar to Minstrel, Skald retrieves and executes author-level plans (ALPs) as part of the story generation process. Ultimately, the system constructs a connected graph with story frames as nodes, as depicted in Table 2. Most commonly, these frames are a trio consisting of a goal which *plans* an act, which, in turn, *intends* a state to occur, and wherein the state ultimately *achieves* the goal. Many of the narratives that Skald generates are formed by combining and connecting similar frame trios.

Despite being an adaptation of the original Minstrel system, Skald follows the same core ideas of simulating the human authoring process. For this reason, Skald is a suitable creative narrative generator to formalize with SIGs because it represents a valid model of computational creativity and is openly available for development. We claim that SIGs

Table 2 An example narrative generated by Skald ('Story A'). The story frames have been manually ordered and translated into natural text for readability. Each frame is composed of symbols, which may be empty, unknown, or contain a specified value.

Natural Language Equivalent	Story Frame
Frederick, the knight, did not want to be injured.	stayhealthy -> (goal) Map(*actor* -> Frederick(Knight), *object* -> Frederick(Knight), *scale* -> <empty slot>, *to* -> <empty slot>, *type* -> "Healthy", *value* -> <empty slot>)
But Fafnir, a dragon, hated Frederick.	hates -> (state) Map(*actor* -> Fafnir(Dragon), *object* -> <empty slot>, *scale* -> "Strong", *to* -> Frederick(Knight), *type* -> "Affect", *value* -> "Negative")
So, Fafnir wanted to injure him.	wantinjure -> (goal) Map(*actor* -> Fafnir(Dragon), *object* -> Frederick(Knight), *scale* -> <empty slot>, *to* -> <empty slot>, *type* -> "C-Health", *value* -> "Injured")
He fought Frederick by blowing a magical flame at him.	attack -> (act) Map(*actor* -> Fafnir(Dragon), *from* -> <empty slot>, *object* -> Flame(Magic), *to* -> Frederick(Knight), *type* -> "Fight")
Frederick was injured by the flame. His plan to stay healthy had been thwarted by Fafnir the Dragon.	injured -> (state) Map(*actor* -> Frederick(Knight), *object* -> <empty slot>, *scale* -> <empty slot>, *to* -> <empty slot>, *type* -> "Health", *value* -> "Injured")

are appropriate for three reasons, namely, they (1) provide a formal representation that can facilitate comparison between story generators beyond Skald, (2) are a bridge towards improved natural language generation in Skald and other generators, (3) expand the library of Skald without additional manual authoring.

3.3 The Story Intention Graph as a Formalism for Imaginative Recall

The SIG model provides formal, concise, and expressive [5] representations for computer-generated narratives. A shared, growing corpus of over one hundred encodings is currently available to describe and investigate narrative structures. By translating stories into SIG encodings, we have a means of expressing the diversity of structures and relationships that can be created by automated narrative generators. The discourse relations defined by SIGs

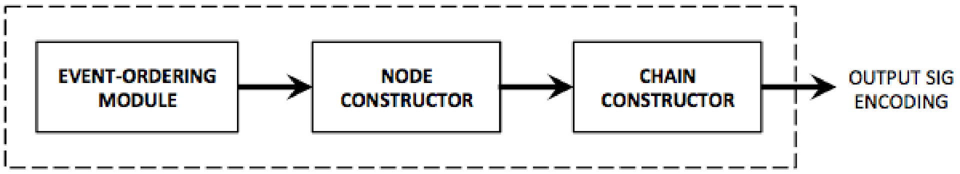


Figure 1 Block diagram of a Skald-to-SIG conversion system.

are useful in corpus annotation as well as algorithmic treatment, particularly related to analogical reasoning. A key aspect of case-based reasoning systems is the distance function used to identify similar cases during the recall phase. Current CBR-based story generators take a parameterized generalization of situations and compute a direct frame comparison to recall cases. To scale such a representation requires significant addition of semantic information to case frames, including a richer distance function to find appropriate cases from the library. Further, the transformation processes mostly generalize at the level of a single parameter's domain constraints. It has been shown [4] that the SIG formalism outperforms other representations in finding not only analogical stories individually, but also analogical sub-sets through a comparison on isomorphic sub-graphs to common SIG patterns.

The SIG model is an encoding of narrative that forms a *semantic network*. Such networks are commonly utilized in cognitive psychology for narrative comprehension studies with humans [7]. In plan-based narrative generation systems, such encodings are used within representations of plan operators and heuristic functions to search for stories [2, 1, 12]. In work related to common sense reasoning from narratives, the predominant representation has been first-order logic [8, 11]. Recent work on statistical mining of narratives [6, 14] strives to find narrative patterns from large web-corpora. Rishes et al. have proposed an automatic method for converting between the Story Intention Graph (SIG) representation to a natural language generator such as PERSONAGE [13].

The process that Skald undergoes is analogous to that of a human storyteller, in that the system considers and modifies past story examples. However, Skald generates a graph representing a bare plotline as its output, and this representation is insufficient for more rich and complex narratives. Thus far, SIGs have only been applied as an analytical tool on pre-written stories with simple plot structures and character attributes. However, SIGs have the potential to express a richer set of stories when combined with a sufficiently creative generator. Once a narrative is represented in terms of SIGs, we can then transform the story with these SIG representations to result in creative retellings.

4 Translating Generated Plotlines into SIGs

We have developed a system that takes in Skald story data as input and produces SIG encodings. Figure 1 shows a block diagram that details the main steps of the procedure, and the following sections will describe each component of the system in detail.

4.1 Event Ordering

Skald generates a story graph without always indicating the ordering of frames. While not every narrative generation system may require event ordering, we included a module for this purpose so that any story generated by Skald will be told in the proper sequence.

Table 3 An example that demonstrates how frames from Story A are sorted by the EOM.

Sorting Step	Order of Events
1	t1: attack -*intends*- injured, t2: hates -*motivates*- wantinjure, t3: injured -*thwarts*- stayhealthy, t4: wantinjure -*plans*- attack
2	t1: attack -*intends*- injured, t2: injured -*thwarts*- stayhealthy, t3: hates -*motivates*- wantinjure, t4: wantinjure -*plans*- attack
3	t1: hates -*motivates*- wantinjure, t2: wantinjure -*plans*- attack, t3: attack -*intends*- injured, t4: injured -*thwarts*- stayhealthy
4	t1: hates -*motivates*- wantinjure, t2: wantinjure -*plans*- attack, t3: attack -*intends*- injured, t4: injured -*thwarts*- stayhealthy
5	t1: hates -*motivates*- wantinjure, t2: wantinjure -*plans*- attack, t3: attack -*intends*- injured, t4: injured -*thwarts*- stayhealthy

While frames generated by the original Skald system are not ordered in the natural language telling, their implied ordering may be discerned by examining the graph connections between events. We define a *frame pairing* as a set of two frames generated by Skald, wherein one directly connects to the second. For instance, Fafnir attacking Frederick in Story A is connected to his intention to injure him by an *intends* link. In this example, the attacking action intends the injured state, and *attack* and *injured* are a pair.

The Event-Ordering Module (EOM) works as follows: for each frame-consequence pairing, search for the given consequence in the remaining events. If the frame is found, swap the found frame to directly follow the current pairing; then, continue reading through the list. If the frame is not found, move the lines succeeding the current line to the head of the list of frame-consequence pairings; then, begin reading again from the beginning. If not found last, the frame with a consequence matching the final frame is tagged so the module does not check the final two pairings, which should be already sorted.

4.2 Node Construction

In accordance with Elson [4], the Node Constructor (NC) unit categorizes each story element as a Proposition (P), Goal (G), or Belief (B) node. Skald already labels frames as states, goals, and actions, which simplifies the conversion process. Every element of the output graph must then be translated into a discourse relation and annotated with the correct agents, objects, and any other related entities as defined by Elson [4]. Because Beliefs and Goals are frames containing content, they are labeled and filled with one or more Interpretive Proposition (I) relations. In Skald, the affectual impact of a P node or actualized I node is merely implied with frame-consequence pairings and whether goals are achieved. To create a proper SIG encoding, Affectual (A) nodes are created for each character of the story.

Table 4 An example narrative generated by Skald ('Story A'). The story events have been manually ordered and translated into natural text for readability.

Order (t)	Node	Links
1	P: injured(Frederick, False)	*actualizes* (t2)

2	G (Frederick):
	injured(Frederick, False)
		provides for A: Frederick

3	G (Fafnir):
	harm(Fafnir, Frederick)
		provides for A: Fafnir;
		damages A: Frederick

4	P: attack(Fafnir, Frederick)	*actualizes* (t3)
5	P: injured(Frederick, True)	*ceases* (t2)

4.3 Chain Construction

Once all nodes are established, they must be linked to complete the SIG encoding process. This process is ensured by the Chain Constructor (CC) module, which reviews the given frame-consequence pairings to make decisions about how P and I nodes (including Goals and Beliefs) are linked. For instance, consider the original pairing of 'wantinjure -*plans*-attack' in Story A. In this case, *wantinjure* is classified as a Goal, and *attack* is known to be a P node that takes place in at t=4. Fafnir deciding to attack Frederick, then, at least *attempts to cause* the state of Frederick becoming injured. The attack also intends and results in Frederick becoming injured at t=5, which thwarts his plan to stay healthy. Consequently, a *ceases* link is established between Frederick's goal to stay healthy, and the P node representing the attack in the story. Notably, the previous *attempt to cause* link is changed to become *actualizes*, as Fafnir succeeded in his goal of injuring Frederick.

The system connects each I node to corresponding A nodes by considering the effects of that I on each agent's goals. If a goal is met for an agent when an I node is carried out, a *provides-for* link is established between an agent and that node. Conversely, a *damages* link is created when the current I node thwarts an agent's goal. If any A nodes contain no links by the end of the chain construction process, they are removed from the final graph.

4.4 Output Visualization

At present, our system outputs text that describes a graph structure representing the SIG encodings; Table 4 conveys this information. An example of how this graph would be represented using Story A and Elson's timeline format is shown in Figure 2, while a second story (Story B) is shown in Figure 3.

5 Perspectives and Future Work

By providing Skald with a SIG case library and specifying rules for SIG-based transformations, we can apply the TRAM procedure to the SIGs themselves. For instance, Story A matches the 'Goal (Desire to Harm)' SIG pattern. By instructing Skald to examine the underlying components of the SIG, and searching for similar patterns, the elements of the original story are then adapted for use in a new SIG template. Thus, when transforming Story A, multiple new stories should be produced. For instance, our modified version of Skald could use a GeneralizeLink TRAM template to recognize that the *actualizes* link at t4 can be replaced with an *attempt to cause* link. An *actualizes* link is then created between t4 and a new I node which represents the opposite of the *injures* action ('heals'). Based on the original

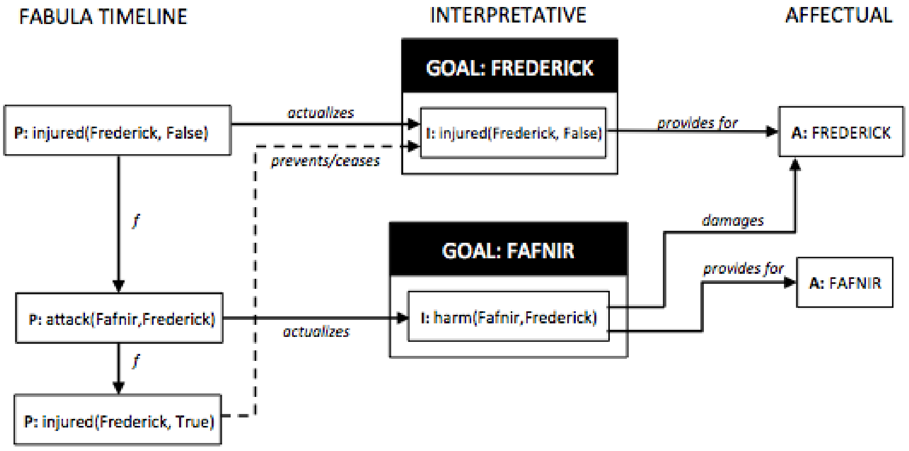


Figure 2 A visual example of the completed SIG encoding for Story A. Story A ultimately follows the 'Goal (Desire to Harm)' SIG pattern.

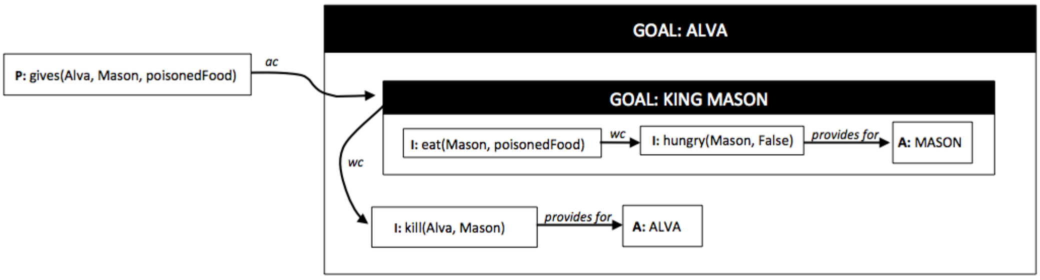


Figure 3 A SIG encoding derived from a second story generated by Skald ("Story B"). Story B includes nested goals and follows the 'Hidden Agenda' pattern. In Story B, a witch named Alva wants to kill King Mason by giving him poisoned food. Mason is hungry, and so accepts the food. Both Alva and Mason's goals are achieved; however, Mason dies by the end of the story.

narrative constraints, the system understands that Frederick being healed is consistent with his goals and thwarts Fafnir's goals, leading to the appropriate connections between the A nodes. The final state, Frederick not being injured, is updated based on the new I node. However, because this state was already a part of the timeline (t1), the final state is removed from the graph, and Frederick's goal by the end of the story is achieved. The resulting story follows the 'Unintended Aid' SIG pattern (Figure 4).

6 Conclusion

We have prepared Skald for improved natural language generation by (1) ordering the frames it produces in graph form, and (2) encoding the story events with story intention graphs. Further, we have extended Skald as a creative system by adding SIGs as a second means of transforming generated stories. Rather than having independent architectures with distinct ways of implementing narrative structure, we can generate more complex stories by working from the SIG specification directly. Output text of other generators may be re-encoded as SIGs, thus enabling comparison between different story generation systems.

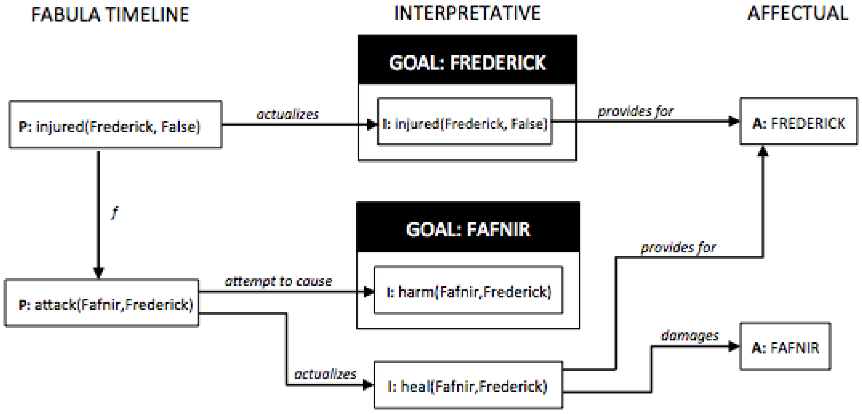


Figure 4 The visual SIG encoding for Story A, when transformed by a modified version of Skald.

The SIG representation, and others like it, enable the expansion of surface realization as an expressive medium. This is true even when the general plots are predictable, implying that stories may be improved even with the same knowledge structures. Future research should work towards quantifying this improvement, as well as to further increase the creative capacity of narrative systems. Future research could also work towards applying the SIG translation process to creative narrative generators beyond Skald, and analyzing variations in the types and diversity of SIG encodings they are able to produce.

Acknowledgements. Sarah Harmon was supported through the BSOE seed funding program. We would also like to thank Brandon Tearse for the development of Skald, Peter Mawhorter for his assistance in reimplementation, and Noah Wardrip-Fruin for his feedback on SIG representation.

References

1 Byung-Chull Bae and R. Michael Young. A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach. *Interactive Storytelling*, 4:156–167, 2008.

2 Yun-Gyung Cheong and R. Michael Young. Narrative generation for suspense: Modeling and evaluation. *Interactive Storytelling*, 4:144–155, 2008.

3 Natalie Dehn. Story generation after TALE-SPIN. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 16–18, 1981.

4 David K. Elson. Detecting story analogies from annotations of time, action and agency. In *Proceedings of the LREC 2012 Workshop on Computational Models of Narrative*, Istanbul, Turkey, 2012a.

5 David K. Elson. Dramabank: Annotating agency in narrative discourse. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, Turkey, 2012b.

6 Andrew S. Gordon and Reid Swanson. Identifying personal stories in millions of weblog entries. In *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*, San Jose, California, 2009.

7 Arthur C. Graesser, Kathy L. Lang, and Richard M. Roberts. Question answering in the context of stories. *Journal of Experimental Psychology: General*, 120:254–277, 1991.

8 Jerry R. Hobbs and Andrew S. Gordon. Encoding knowledge of commonsense psychology. In *Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning*, pages 107–114, Corfu, Greece, 2005.

9 Sheldon Klein, John F. Aeschlimann, David F. Balsiger, Steve L. Converse, Claudine Court, Mark Foster, Robin Lawo, John D. Oakley, and Joel Smith. Automatic novel writing: A status report. Technical report 186, Computer Science Department, The University of Wisconsin, Madison, 1973.

10 James R. Meehan. Tale-spin, an interactive program that writes stories. In *Proceedings of the fifth International Joint Conference on Artificial Intelligence (IJCAI'77)*, volume 1, pages 91–98, Cambridge, MA, 1977. San Mateo, CA: Morgan Kaufmann.

11 Erik T. Mueller. Modelling space and time in narratives about restaurants. In *Literary and Linguistic Computing*, volume 4, 2006.

12 James Niehaus and R. Michael Young. A computational model of inferencing in narrative. *AAAI Spring Symposium: Intelligent Narrative Technologies II*, 2009.

13 Elena Rishes, Stephanie M. Lukin, David K. Elson, and Marilyn A. Walker. Generating different story tellings from semantic representations of narrative. In *Koenitz, H., Sezen, T.I., Ferri, G., Haahr, M., Sezen, D., C atak, G. (eds.) ICIDS 2013, LNCS*, volume 8230, pages 192–204. Springer, Heidelberg, 2013.

14 Reid Swanson and Arnav Jhala. A crowd-sourced collection of narratives for studying conflict. In *Language Resources and Evaluation Conference Workshop on Computational Models of Narrative (LREC 2012)*, Istanbul, Turkey, 2012.

15 Brandon Tearse. Minstrel Remixed and Skald, 2015. University of California, Santa Cruz, 2012. Web.

16 Brandon Tearse, Peter Mawhorter, Michael Mateas, and Noah Wardrip-Fruin. Skald: Minstrel reconstructed. *IEEE Transactions on Computational Intelligence and AI in Games*, 6:156–165, 2014.

17 Scott Turner. Minstrel: a computer model of creativity and storytelling. Technical Report CSD-920057, Ph.D. Thesis, Computer Science Department, University of California, Los Angeles, CA, 1992.

Animacy Detection in Stories

Folgert Karsdorp[1], Marten van der Meulen[1], Theo Meder[1], and Antal van den Bosch[2]

1 Meertens Institute
Amsterdam, The Netherlands
{folgert.karsdorp,marten.van.der.meulen,theo.meder}@meertens.knaw.nl

2 Radboud University
Nijmegen, The Netherlands
a.vandenbosch@let.ru.nl

Abstract

This paper presents a linguistically uninformed computational model for animacy classification. The model makes use of word *n*-grams in combination with lower dimensional word embedding representations that are learned from a web-scale corpus. We compare the model to a number of linguistically informed models that use features such as dependency tags and show competitive results. We apply our animacy classifier to a large collection of Dutch folktales to obtain a list of all characters in the stories. We then draw a semantic map of all automatically extracted characters which provides a unique entrance point to the collection.

1998 ACM Subject Classification I.2.7 Natural Language Processing

Keywords and phrases animacy detection, word embeddings, folktales

Digital Object Identifier 10.4230/OASIcs.CMN.2015.82

1 Introduction

For almost all species in the world, the capacity to distinguish animate objects from inanimate objects is essential to their survival. Those objects could be prey, for example, or predators, or mates. The fundamental nature that the distinction between animate and inanimate has for humans is reflected in the fact that this division is acquired very early in life: children of less than six months old are well able to distinguish the two categories from one another [16]. Moreover, recent brain research shows that the distinction appears in the organization of the brain (e.g. [8]). For some researchers, this provides evidence for the idea that the division between animate and inanimate is an innate part of how we see the world.

Although animacy may be a scalar rather than a strictly categorical distinction (see e.g. the animacy hierarchy in [4] and research such as [25]), the animate/inanimate distinction is traditionally taken as binary with regard to lexical items: something is either animate (e.g. a human) or not (e.g. a chair). This standpoint has been challenged, however, by researchers from different fields. Firstly, it has long been established in linguistic typology that not all languages award animacy to the same entities in different grammatical categories. As [4] notes, many languages, such as, for example, English, distinguish between human and not-human in the choice of pronouns; other languages, such as Russian, distinguish between animate (entailing humans and animals) versus non-animate (entailing everything else) in their interrogative pronouns. This indicates different subdivisions of animacy in the respective languages. Secondly, philosophers such as Daniel Dennett support the view that animacy and aliveness are to be treated as epistemological stances rather than fixed states in the world: not ineffable qualia but behavioral capacity defines our stance towards objects [6].

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 82–97
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In other words, depending on whether people *think* that an object is animate, they utilize different cognitive strategies to explain and predict the actions of those objects. Finally, evidence from psycholinguistic research has accumulated to support this view of animacy as a cognitive viewpoint rather than an extra-perceptive absolute. Nieuwland & Berkum [15] for example show that college student test subjects readily accept animate behavior from inanimate objects within the proper contexts, and Vogels et al. [9] moreover emphasize the relation between animacy and motion, showing that factors such as self-propelment play a crucial role in recognizing or awarding animacy to certain objects. This is exemplified in the opening of this well-known story:[1]

> *A farmer bought a pancake on the market. Once he got home, the farmer was hungry and began to bake the pancake. The farmer tried one of his skillful flipping techniques, but he failed and the pancake fell on the ground. Coincidentally, the door of the kitchen was open and the pancake rolled out to the field, as hard as he could...*

Although initially, based on their knowledge of the world, readers will regard the pancake as inanimate, the self-propelled motion verb 'rolled' initiates our shift towards an animate interpretation of the pancake. As readers (or listeners) of a story, we choose to view participating objects at varying levels of abstraction in order to predict their behavior. Dennett [6] defines three levels of abstraction: (1) the *physical stance*, (2) the *design stance* and (3) the *intentional stance*. The physical stance deals with predictions about objects given their physical properties. The design stance deals with concepts such as purpose, function or design. The intentional stance is concerned with belief, thinking and intentions. These are all cognitive strategies we use to predict and explain the actions of objects in our environment. Interestingly, in the process of reading the opening of the story about the fleeing pancake, readers and listeners experience the transition from one strategy to the next quite clearly. Initially, the pancake is interpreted from a physical stance, or perhaps the more abstract design stance in terms of the purpose (i.e. to stave off hunger). It is only at the last adverbial phrase 'as hard as he could' that we start to wonder whether we should adopt to the yet more abstract intentional stance and consider the pancake to be a rational agent.

Given the fundamental nature of the distinction between animate and inanimate, it is perhaps not too surprising that it has proven to be useful in a variety of natural language processing tasks dealing with e.g. anaphora resolution and dependency parsing [18, 11, 22]. Existing methods for the automatic labeling of text for animacy are usually rule-based, machine-learning-based, or a hybrid of these methods. Common to most approaches is the fact that they make use of semantic lexicons with information about animacy, as well as syntactic cues in a text. Both feature types are relatively costly to obtain as they require lexical resources or syntactic parsing systems, which, with the exception of a few languages, are not readily available.

In this paper we present a new linguistically uninformed model to automatically label texts for animacy. We show that we can do away with features that require syntactic parsing or semantic lexicons while still yielding competitive performance. We focus on labeling animacy in stories because stories pose some particularly interesting problems to automatic systems of animacy recognition. As the example of the fleeing pancake already illustrated, in stories any entity may at some point exhibit animate behavior, even when they are inanimate in the 'real' world. Another example is the *Sorcerer's Apprentice* sequence in Walt Disney's

[1] `http://www.verhalenbank.nl/items/show/9636`

famous *Fantasia*, in which brooms display the ability to collect buckets of water. Such examples, where pancakes, brooms and other entities act as animate beings, make a clear case for developing dynamic, data driven systems that do not rely too much on static and fixed world knowledge, but rather on immediate context.

The remainder of this paper is structured as follows. We will start with a short overview of existing techniques for automatically labeling animacy in texts, including the definitions of animacy used in these papers (§2). After a description of the corpus used in our study and how the annotations of the corpus have been established (§3), we will give an account of our computational models in Section 4. We report on the empirical results in Section 5. Next, we provide an evaluation on a larger dataset, while also showing a real-world application of our animacy detection system (§6). The final section offers our conclusions and possible directions for future research.

2 Previous Work

A handful of papers deal with automatic animacy detection. Most approaches make use of rule-based systems or machine learning systems with morphological and syntactic features. [7] present a rule-based system that makes use of the lexical-semantic database WordNet. They label each synset in WordNet for animacy. Using a variety of rules to detect the head of an NP, they use the fraction of synsets in which a particular noun occurs to arrive at a classification for animacy. [17] extend their previous algorithm by first determining the animacy of senses from WordNet on the basis of an annotated corpus. They then apply a k-nearest neighbor classifier using a number of lexical and syntactic features alongside features derived from WordNet to arrive at a final animacy classification.

[19, 20, 21] present a number of animacy classifiers that make use of syntactic and morphological features. These features include the frequency of analysis of the noun as 'subject' or 'object', the frequency of the occurrence of a noun in a passive *by*-phrase, and the frequency of the noun as a subject followed by either animate personal pronouns or inanimate personal pronouns. These features are then aggregated for each lemma after which a machine learning system (decision tree or k-nearest neighbor classifier) is trained. A similar approach is presented in [3]. In this study a Maximum Entropy classifier is trained on the basis of three feature types: (1) bag-of-words with and without their corresponding Part-of-Speech tags, (2) internal syntactic features such as the syntactic head and (3) external syntactic features that describe the dependency relation of a noun to a verb (i.e. subject relation, object relation etc.) This is the only study that makes use of a corpus fully labeled for animacy. In an approach partially related to animacy detection, [10] attempt to extract the cast (i.e. all characters) from a story. Similar to [3] they rely on dependency tags to extract the subjects of direct and indirect speech.

[1] present a model that attempts to generalize the animacy information in a lexical-semantic database of Dutch by augmenting 'non-ambiguous' animate entries with contextual information from a large treebank of Dutch. They apply a k-nearest neighbor algorithm with distributional lexical features that aim to capture the association between a verb or adjective and a particular noun. The idea is that nouns that occur in similar contexts as animate nouns are more likely to be animate than nouns that occur more frequently in contexts similar to inanimate nouns.

[14] present an approach that combines a number of animacy classifiers in a voting scheme and aims at an interpretable and correctable model of animacy classification. A variety of classifiers is used, such as the WordNet-based approach of [7], named entity recognition systems, and dictionary sources.

The approaches mentioned above present us with a number of problems. First, nearly all of them rely heavily on costly, linguistically informed features derived from lexical-semantic databases or syntactic parsing. For most languages in the world, however, we cannot rely on these resources, either because they do not exist, or because their performance is insufficient. Second, animacy detection is often seen as a useful feature for a range of natural language processing techniques, such as anaphora resolution and syntactic parsing. The mutual dependence between these techniques and animacy detection, however, is in fact a chicken-and-egg situation.

Another major problem with the approaches above is, as said earlier, that they are lemma-based, which means that the models are generally insensitive to different usages of a word in particular contexts. In other words, in most of the literature on automatic animacy detection, a static, binary distinction is made between animate and inanimate. [3] for example, define objects as animate if they are alive and have the ability to move under their own will. [18] define animacy in the context of anaphora resolution: something is animate "if its referent can also be referred to using one of the pronouns he, she, him, her, his, hers, himself, herself, or a combination of such pronouns (e.g. his/her)". However, as was explained above, these definitions are not necessarily in line with current linguistic and neurological research [15]. Similarly, they are not particularly applicable to the rich and wondrous entities that live in the realm of stories. As was shown above, although a pancake is typically not an animate entity, its animacy depends on the story in which it appears, and even within the story the animacy may change. To accommodate this possibility, we therefore choose to define animacy in terms of Dennett's intentional stance, which is more dynamic, and which ultimately comes down to the question whether "you decide to treat the object whose behavior is to be predicted as a rational agent" [6, pp. 17]. Our system for animacy detection therefore needs to be dynamic, data driven, and token-based. It may to some extent rely, but cannot rely too heavily, on static world knowledge.

3 Data, Annotation and Preprocessing

To develop this dynamic data-driven system we use a corpus of Dutch folktales. As argued in the introduction, our reason to use folktales is that, as [9] note, 'In cartoons or fairy tales [...] inanimate entities or animals are often anthropomorphized', which means that the material could yield interesting cases of unexpected animacy, as is the case with the pancake in *The fleeing pancake* and the broomsticks in *Fantasia*.

Our initial corpus consists of 74 Dutch stories from the collection *Volkssprookjes uit Nederland en Vlaanderen*, compiled by [27]. The collection is composed of Dutch and Flemish retellings of popular and widespread stories, including such tales as *The Bremen Town Musicians* (ATU 130) [2] and *The Table, the Ass, and the Stick* (ATU 563), as well as lesser-known stories such as *The Singing Bone* (ATU 780) and *Cock, Hen, Duck, Pin, and Needle on a Journey* (ATU 210). This last story is again a clear example where otherwise inanimate objects are animated, as it concerns the adventures of several household items, such as a *pin*, a *hackle*, an *egg*, and a *whetstone*. A digital version of the collection is available in the Dutch Folktale Database from the Meertens Institute (corpus SINVSUNV.20E).[3] Using a single collection for our corpus presents us with a helpful homogeneity with regard

[2] The ATU numbers refer to the classificatory system for folklore tales, as designed by Aarne, Uther and Thompson [28].

[3] See `http://www.verhalenbank.nl`

to the editor, length of the stories, and language use, as well as exhibiting some content-wise diversity among the collection, which contains fairytales and legends.

All together, the corpus consists of 74,504 words, from 5,549 unique words. Using the annotation tool brat (brat rapid annotation tool), an online environment for collaborative editing[4], two annotators labeled words for animacy, within the context of the story.[5] All unlabeled words were implicitly considered to be inanimate. The following sentence provides an example annotation.

(1) Jij smid, jij bent de sterkste; hou je vast aan de bovenste
ANIMATE ANIMATE ANIMATE ANIMATE
takken, en dan ga jij kleermaker aan zijn benen hangen en zo gaan
ANIMATE ANIMATE ANIMATE
we maar door
ANIMATE
'You, blacksmith, you are the strongest; hold on to the upper branches and then, you, tailor, will grab his legs and so we go on... '

Because we interpreted animacy within the context of the story, the same lexical item could be labeled differently in different stories. For example, in the above-mentioned example of the pancake, which occurs in SINVS076 in our corpus, the pancake is tagged consistently as 'animate'. In another story, SINVS042, where at one point a soldier is baking pancakes, the pancakes do not act, and are thus not labeled as 'animate'. The following sentences show how this was employed in practice.

(2) Terwijl hij de pannekoek bakte, keek hij naar het ding, dat uit de
ANIMATE ANIMATE
schouw gevallen was

'While he was baking the pancake, he looked at the thing, which had fallen from the hearth... '

(3) Toevallig stond de deur van de keuken open en de pannekoek rolde naar buiten,
ANIMATE
het veld in, zo hard hij maar kon.
ANIMATE
'Coincidentally the door of the kitchen was open and the pancake rolled outside, into the field, as fast as it could'

This annotation resulted in 11,542 animate tokens of 743 word types, while implicitly yielding 62,926 inanimate tokens from 5,011 unique inanimate words. Because of our context-dependent approach, some words, such as *pancake* and *egg*, occurred in both animate types as inanimate types, because they were labeled as both animate and inanimate in some cases in our corpus. It is telling that of the animate tokens 4,627 (40%) were nouns and proper nouns, while only 6,878 of the inanimate tokens (11%) are nouns. This shows that being a noun is already somewhat of an indication for animacy. After tokenization with the tokenization module of the Python software package Pattern [5] we fed all stories to the state of the art

[4] `http://brat.nlplab.org`

[5] On the basis of five stories that were annotated by both annotators we computed an inter-annotator agreement score (Cohen's Kappa) of $K = 0.95$.

syntactic parser for Dutch, Alpino [2]. From the resulting syntactic parses, we extracted the features for the linguistically informed models, see Section 4.3.

4 Experimental Setup

This section describes our experimental setup including the features used, the machine learning models we applied, and our methods of evaluation.[6].

4.1 Task description

We formulate the problem of animacy detection as a classification problem where the goal is to assign a label at word level, rather than at lemma level. This label indicates whether the word is classified as animate or inanimate.

4.2 Evaluation

Inanimate words far outnumber animate words in our collection (see §3). Reporting accuracy scores would therefore provide skewed results, favoring the majority category. The relative rarity of animate words makes evaluation measures such as the well-known $F1$-score more appropriate. For this reason, we report on the precision, recall and $F1$-score [30] of both classes for all experiments. Also, while in most of the literature on animacy detection results are only presented for the classification of nouns or noun phrases, we will, while reporting on nouns and noun phrases as well, additionally report on the results for all words in a text.

In real-world applications an animacy detection system will most likely be faced with completely new texts instead of single words. It is therefore important to construct a training and test procedure in such a way that it mimics this situation as closely as possible. If we would, for example, make a random split of 80% of the data for training and 20% for testing on the word level, we run the risk of mixing training data with test data, thereby making it too easy for a system to rely on words it has seen from the same text. [3] fall into this trap by making a random split in their data on the sentence level. In such a setup, it is highly likely that sentences from the same document are present in both the training data and the test data, making their evaluation unrealistic. To circumvent this problem, we split the data at the story level. We make use of 10-fold cross-validation. We shuffle all stories, partition them in ten portions of equal size. In ten iterations, each partition acts as a test set, and the other nine partitions are concatenated to form the training set.

4.3 Features

We explore a range of different features and feature combinations including lexical features, morphological features, syntactic features, and semantic features.

4.3.1 Lexical features

We take a sliding-window approach where for each focus word (i.e. the word for which we want to predict whether it is animate or not) we extract both n words to the left and n words to the right, as well as the focus word itself. In all experiments we set n to 3. In

[6] The data set and the code to perform the experiments are available from `https://fbkarsdorp.github.io/animacy-detection`

addition to the word forms, for each word in a window we also extract its lemma as provided by the output of the syntactic parser Alpino.

4.3.2 Morphological Features

For each word we extract its part-of-speech tag. For reasons of comparability we choose to use the tags as provided by Alpino, instead of a more specialized part-of-speech tagger. Again, we take a sliding window approach and extract the part-of-speech tags for three words left and right of the focus word, as well as the tag of the focus word itself.

4.3.3 Syntactic Features

We extract the dependency tag for each word and its $n = 3$ neighbors to the right and to the left as provided by the syntactic parser Alpino. Animate entities tend to take the position of subject or object in a sentence which is why this feature is expected and has proven to perform rather well.

4.3.4 Semantic Features

The most innovative feature we have included in our model is concerned with semantic similarity. In his *Philophische Untersuchungen* Wittgenstein already suggests that “Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache”[7] (PI 43). This is reflected by the well-known insight in computational linguistics that the meaning of words can be approximated by comparing the linguistic contexts in which words appear. In other words: words that often co-appear with the same set of words, will have a more similar meaning. Recently, there has been a lot of interest in procedures that can automatically induce so-called ‘word embeddings’ from large, unannotated collections of texts (e.g. [13, 24]). These models typically attempt to learn vector representation with less dimensions than the vocabulary size for each word in the vocabulary which captures the typical co-occurrence patterns of a word in the corpus. The similarity between words can then be approximated by applying similarity metrics, such as the cosine metric, to these vectors of word embeddings.

We have trained word embeddings with 300 dimensions using the popular skip-gram architecture [13] on the Dutch corpus of COW (COrpora from the Web). COW is a collection of linguistically processed web corpora for English, Dutch, Spanish, French, Swedish and German [26]. The 2014 Dutch corpus contains 6.8 billion word tokens. The idea behind using the word embeddings is that similarities between animate words can be estimated by inspecting the context in which they occur. From this follows, for example, that the word embeddings of an animate word are more similar to those of other animate words, as opposed to the embeddings of inanimate words.

To give an illustration of this idea, in Figure 1 we depict a two-dimensional Principle Component Analysis (PCA) projection of the 300 dimensional word embedding vectors for a number of typically animate and typically inanimate words. The horizontal gray line in the plot illustrates the separability of the animate and inanimate words in the first dimension of the PCA projection. It is interesting to observe that *ghost* is the one closest to all other inanimate entities. Likewise, words such as *castle*, *house* or *car* are often used in figurative language (metonymy), for example to refer to the people owning or living in the castle. Perhaps this ambiguous animacy position is responsible for their position in the first dimension close to real animate entities.

[7] The meaning of a word is its use in the language.

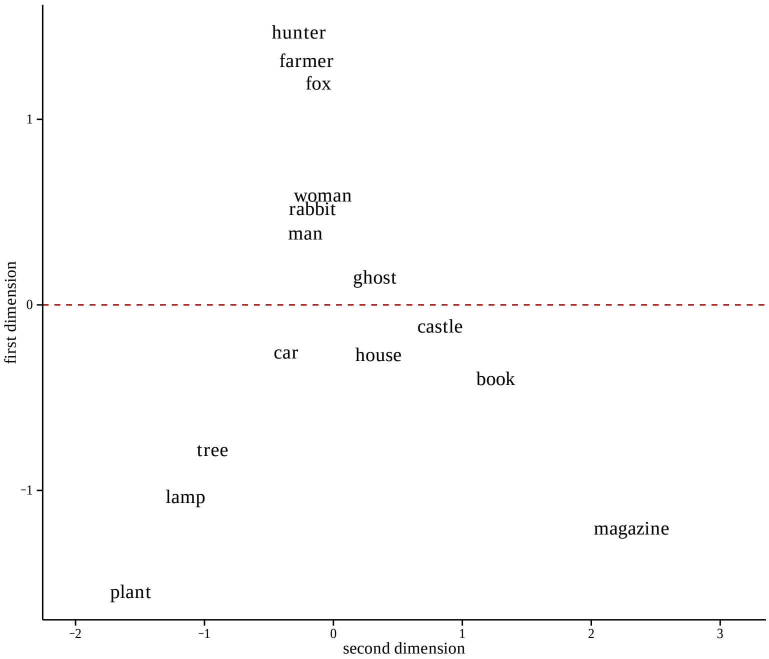


Figure 1 Two-dimensional PCA projection of the 300 dimensional word embedding vectors for a number of animate and inanimate words. The horizontal line illustrates the separability between the two classes in the first dimension.

4.4 Models

We employ a Maximum Entropy classifier with L2 regularization as implemented in [23]. In all experiments, we set the regularization strength parameter C to 1.

We compare nine models in which we make use of different feature combinations: (1) words, (2) words and Part-of-Speech tags, (3) words, Part-of-Speech tags and lemmata, (4) words, Part-of-Speech tags, lemmata and dependency tags, (5) word embeddings and (6-9) the features in model 1 to 4 with word embeddings.

Although our background corpus is sufficiently large to cover most words in an unseen text, there will always be rare words for which we do not have learned word embeddings. Therefore, in order to effectively make use of the word embedding vectors, we need a way to deal with out-of-vocabulary items. We adopt a simple strategy where we make use of a primary classifier and a back-off classifier. For models 6 to 9, we augment each word with its corresponding 300 dimension word embeddings vector. In the case of out-of-vocabulary words, we resort to a back-off model that contains all features except the word embeddings. For example, a model that makes use of words and word embeddings, will make a prediction on the basis of the word features alone. In case of the model that solely uses the embeddings (model 5), the back-off classifier is a majority-vote classifier, which classifies unseen words as inanimate.

5 Results

In Table 1 we present the results for all nine models on the complete data set. For each model we report the precision, recall and $F1$-score for the animate words and the inanimate words.

Table 1 Precision, Recall and $F1$-score for animate and inanimate classes per feature setting for all words.

	inanimate			animate		
	P	R	$F1$	P	R	$F1$
embeddings	0.98	0.99	0.98	0.93	0.89	0.91
word	0.96	0.99	0.98	0.94	0.78	0.85
word + embeddings	0.98	0.99	0.98	0.94	0.90	0.91
word + PoS	0.97	0.99	0.98	0.94	0.86	0.89
word + PoS + embeddings	0.98	0.99	0.99	0.94	0.91	0.93
word + PoS + lemma	0.97	0.99	0.98	0.94	0.86	0.90
word + PoS + lemma + embeddings	0.98	0.99	0.99	0.94	0.91	0.93
word + PoS + lemma + dep	0.97	0.99	0.98	0.94	0.86	0.90
word + PoS + lemma + dep + embeddings	0.98	0.99	0.99	0.94	0.92	0.93

All models perform well on classifying inanimate words. However, since this is the majority class, it is more interesting to compare the performance of the models on the animate instances. It is interesting to observe that the 'simple' n-gram word model already performs rather well. Adding more features, such as Part-of-Speech or lemmata, has a consistently positive impact on the recall of the model, while leaving the precision untouched. As can be observed from the table, employing the rather expensive dependency features shows barely any improvement.

The model that only uses word embedding features is one of the best performing models. This is a context-insensitive model that operates on the level of the vocabulary, which means that it will predict the same outcome for each token of a particular word type. The high precision and high recall show us that this model has acquired knowledge about which words *typically* group with animate words and which with inanimate words. However, the models that combine the word embeddings with the context sensitive features, such as word n-grams or Part-of-Speech tags, attain higher levels of precision than the context-insensitive model. The best performance is achieved by the model that combines the word features, Part-of-Speech tags and the word embeddings. This model has an $F1$-score of 0.93 on animate words and 0.99 on inanimate words. Adding more features does not result in any more performance gain.

Table 2 zooms in on how well nouns and names are classified. The best performance is again achieved by the model that combines the word features with the part-of-speech tags and word embeddings, resulting in an $F1$-score of 0.92 for animate instances and 0.95 for inanimate instances. The relatively lower score for the inanimate class can be explained by the fact that relatively easy instances, such as function words, which are never animate, are not included in the score now.

6 A Semantic Map of Animate Entities in the Dutch Folktale Database

Our approach to animacy classification appears to be successful. In this section we employ our classification system to extract all animate entities from unannotated folktales from the

Table 2 Precision, Recall, and $F1$ score for animate and inanimate classes per feature settings for all words tagged as noun.

	inanimate			animate		
	P	R	$F1$	P	R	$F1$
embeddings	0.90	0.96	0.92	0.93	0.85	0.89
word	0.78	0.98	0.87	0.96	0.60	0.74
word + embeddings	0.90	0.97	0.93	0.95	0.85	0.90
word + PoS	0.86	0.96	0.90	0.93	0.78	0.84
word + PoS + embeddings	0.93	0.96	0.95	0.95	0.90	0.92
word + PoS + lemma	0.87	0.96	0.91	0.94	0.80	0.86
word + PoS + lemma + embeddings	0.93	0.96	0.94	0.95	0.89	0.92
word + PoS + lemma + dep	0.87	0.96	0.91	0.93	0.80	0.86
word + PoS + lemma + dep + embeddings	0.93	0.96	0.95	0.95	0.90	0.92

Dutch Folktale Database, all of which were not used in the previous experiment.[8] The reason for this is twofold. First, it allows us to further our evaluation of the classifier. In a classical evaluation setup – as with our approach – it is general practice to train a computational system on some training data. The performance of the system is then evaluated on a held-out test set. Our annotated corpus contains a reasonably diverse set of stories in terms of genre, yet it is fairly small and rather homogeneous in style. Even though we performed a cross-validation experiment, there is a chance of 'overfitting' to the style of the subset of folktales we trained on. The second reason for applying the classifier to such a large collection is to enrich the collection with a character-based information layer, allowing researchers to browse the collection in new ways.

6.1 Data

For our evaluation we make use of a sub-collection of folktales from the Dutch Folktale Database. The complete collection consists of about 42,000 folktales [12], and contains stories from various genres (e.g. fairytales, legends, urban legends, jokes, personal narratives) in standard Dutch and Frisian, as well as in a number of dialectal variants. Every entry in the database contains meta-data about the story, including language, collector, place and date of narration, keywords, names, and sub-genre. For our paper we make use of a sub-collection comprising 16,294 stories written in standard Dutch. The distribution of genres in the subcollection is the following: urban legends ($n = 2,795$), legends ($n = 299$), jokes ($n = 3,986$), personal narratives ($n = 693$), riddles ($n = 1,626$), sagas ($n = 6,045$) and fairy tales ($n = 832$). We evaluate a random sample of this sub-collection ($n = 212$) in which this genre distribution is taken into account.

6.2 Evaluation

Our definition of animacy allows us to utilize our animacy detection system to extract all characters from a story in a similar vein as [10]. The system labels each noun and name in a text for animacy. After removing duplicate words, this produces a set of words that

[8] `http://www.verhalenbank.nl`

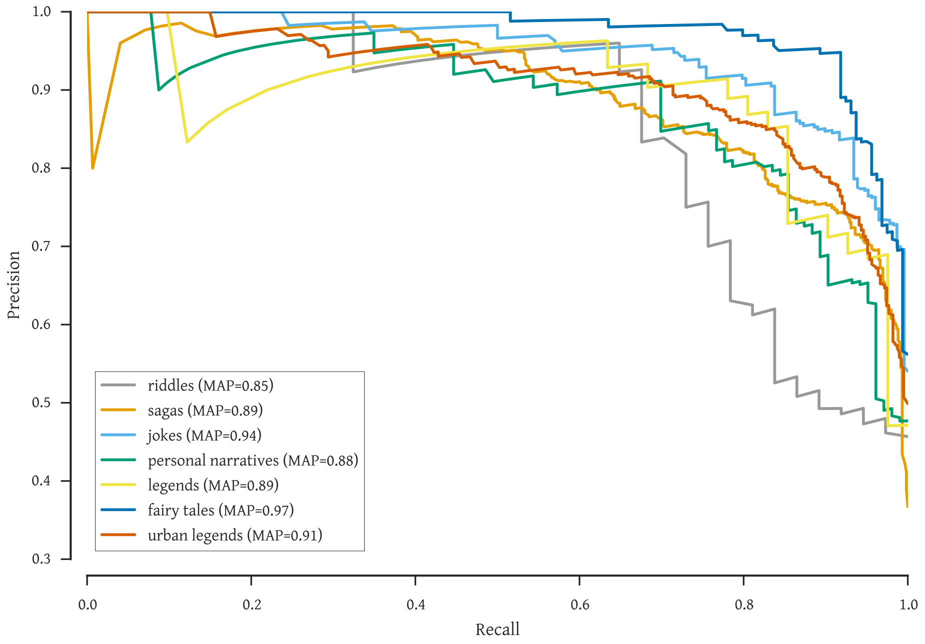


Figure 2 Precision-Recall Curves and Mean Average Precision scores per genre.

comprises the cast of a story. Without gold standard annotations, however, we can only evaluate these character sets for precision and not for recall. An alternative approach is to produce a ranking of all words in a story where the goal is to allocate the highest ranks to animate entities. This allows us to evaluate individual rankings using *Average Precision* which computes the average over precision scores at increasing points of recall. We compute the *Average Precision* as follows:

$$\text{AP} = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{\text{number of relevant items}} \tag{1}$$

where k is the position in the ranked list of n retrieved items. $P(k)$ represents the precision at k and $rel(k) = 1$ if the item at k is relevant, $rel(k) = 0$ otherwise.

Per genre, a Mean Average Precision (MAP) can be computed as the normal average of the AP values of all instances within the genre.

Naturally, with this evaluation method, we still need to manually evaluate the rankings. By using a rank cutoff and evaluating a sample of all automatically annotated stories, we reduce the costly manual labor to a minimum. We order all nouns and names in a story using the output of the probabilistic decision function of the Maximum Entropy classifier. After removing duplicate words, this produces a final ranking. The rankings are evaluated with a rank cutoff at 50.

6.3 Results

We present the results in Figure 2 in which we show the Precision-Recall curve as well as the Mean Average Precision (MAP) score for each genre. The Precision-Recall curve is obtained from computing precision-recall pairs for different probability thresholds. The

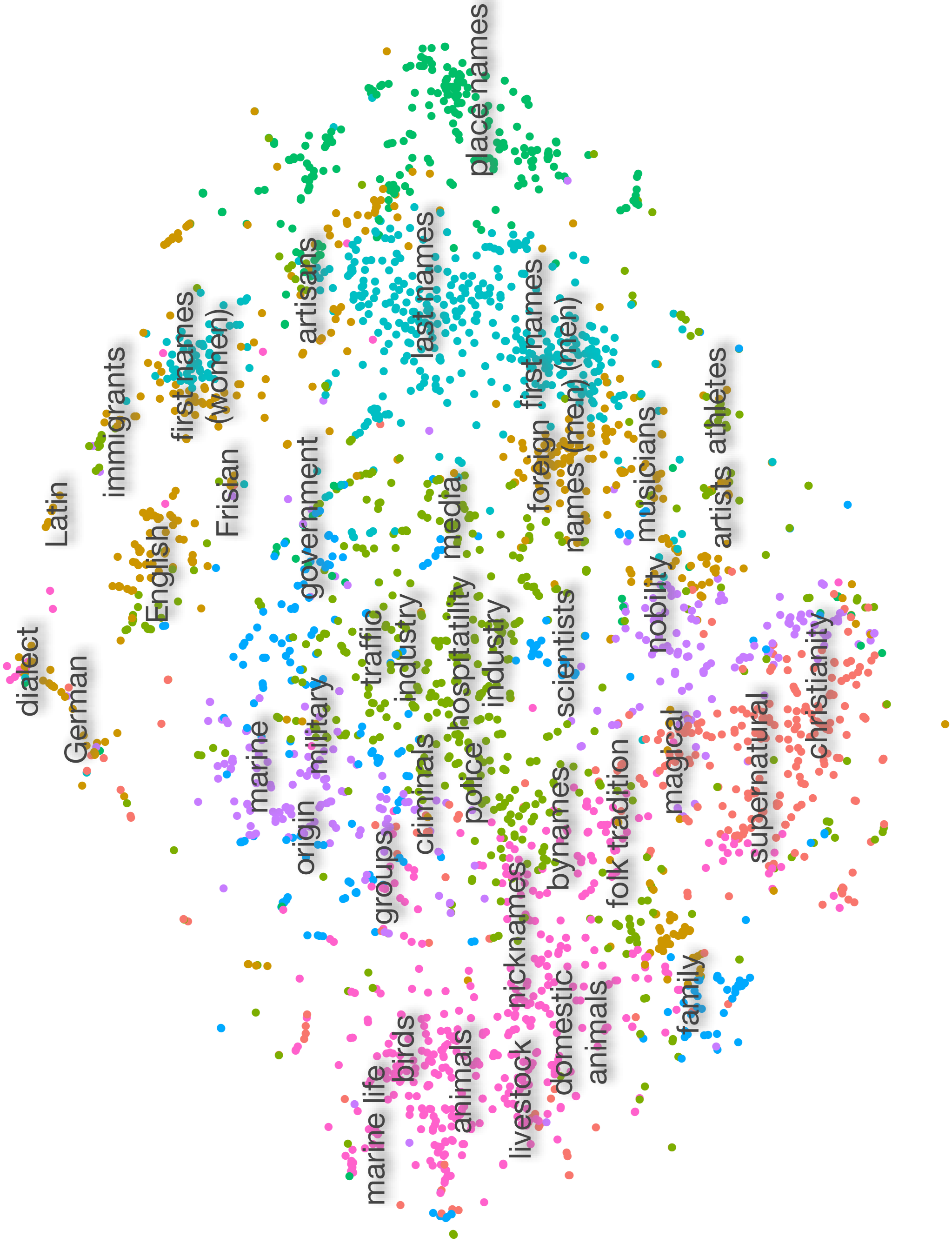


Figure 3 Visualization of characters in the Dutch Folktale Database based on their embeddings using t-SNE.

system performs well, especially on fairytales (MAP= 0.97) and jokes (MAP= 0.94).[9] The lowest performance is measured on riddles (MAP= 0.85). This lower score is partly due to the system's inability to position the word *blondje* ('dumb blond' with a pejorative connotation) high up the ranking.

6.4 A Semantic Map of Characters

The word embeddings that we used as features for our animacy classifier can be employed to describe the similarities and dissimilarities between the extracted animate entities. In Figure 3 we present a two-dimensional semantic map that depicts the (dis)similarities between all extracted animate entities.[10] The dimension reduction was performed using t-Distributed Stochastic Neighbor Embedding (t-SNE) [29]. The coloring of the nodes was obtained by applying a k-Means cluster analysis (k=8) to the word embeddings.

The map discloses a rich diversity of animate entities grouped into semantically coherent clusters. The pink cluster on the far left represents a grouping of all kinds of animals. Note that within this cluster there exist many subtle sub-clusters describing more specific positions in the animal taxonomy, e.g. birds and livestock, marine life, and insects. The central green cluster is occupied by characters of different professions. There is a large number of characters from the hospitality industry, such as waiter and cook, as well as from the transport sector, such as chauffeur and train conductor. One of the interesting groupings is located at the very bottom of the map. This cluster describes magical, supernatural and Christian characters (henceforth supernatural cluster). In Figure 4 we provide a detailed view of this cluster.

The supernatural cluster is noteworthy because it is, like the animal cluster, highly structured. Several clear hierarchically ordered clusters are discernible in Figure 4, with several subgroups emerging. The lower right hand corner for example entails religious or even Christian professions, such as 'bishops' and 'vicar'. From there, a link is made via 'catholics' and 'protestants' to the more general 'believers' and 'followers'. This mini-node bifurcates into two different nodes. Firstly, in the middle-right, a cluster is found containing words designating followers of different religions, such as 'Jew' and 'Muslim', which branches of to the top right node, which is a 'religious fringe' node, containing 'cult', 'satanist' and 'Freemasons'. It is interesting that 'wicca', which might be expected to be clustered in this node, as it also represents an organized semi-religious group, is clustered rather with 'magic' and 'witchcraft' in the upper-left 'magic' cluster.

The other cluster connected to the 'believers' and 'followers'-mini node is structurally complex, starting with such terms as 'people' and 'believers', but also containing, strikingly, 'Allah'. Taking into account that the Christian term 'lord' is clustered elsewhere, with adjectives such as 'compassion' and 'glory', but also with 'persecutors', this means that the two deities are embedded very differently. The cluster then continues through 'Satan' and 'Lucifer' to 'angels' and 'guardian angels'. These words form again a bridge towards more esoteric creatures, such as 'nature spirits', culminating in the far left 'martians' and 'superman'. This cluster is connected to the upper left hand cluster, which contains traditional magical creatures such as 'werewolves' and 'dragons'.

[9] A MAP of 0.97 means that on average, nearly all actual cast members of a folktale are ranked on top, with the first case of a non-animate entity entering the ranking at about rank 5 or 6 on average.

[10] Readers are invited to view an interactive version of the map at the following address: `http://fbkarsdorp.github.io/animacy-detection/`.

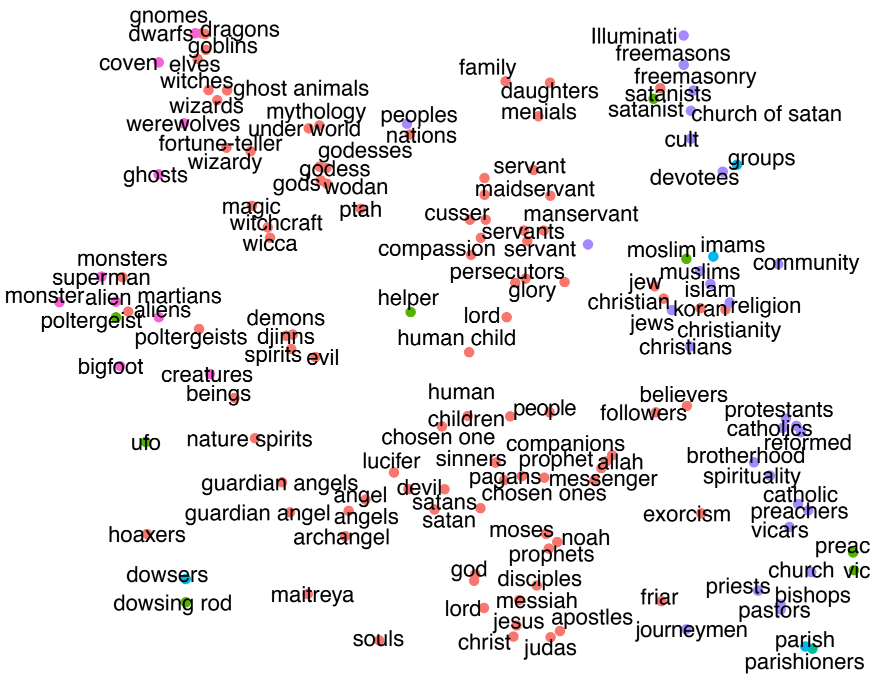


Figure 4 Detailed view of the 'Supernatural' cluster.

In summary, the semantic map makes a case for the successfulness of our approach. The word embeddings combined with the strength of t-SNE to position the characters on a two-dimensional map, yield a powerful representation. The above description is only part of the extremely rich network of associations this semantic map displays.

7 Concluding Remarks

The approach taken in this paper to create a model for animacy classification using linguistically uninformed features proves to be successful. We compared the performance of linguistically informed models (using features such as Part-of-Speech and dependency tags) to models that make use of lower-dimensional representations of the data. With the exception of the model that solely makes use of these representations, all models benefit from adding these features. The model that requires the least linguistic information (word n-grams plus word embeddings) outperforms all linguistically informed models (without embeddings). The best results are reported by the model that combines word n-grams with Part-of-Speech n-grams and word embeddings.

We have the following recommendation for future research. Natural language processing models such as co-reference resolution or linguistic parsing could benefit from a module that filters animate from inanimate candidate words. Since these models typically depend on linguistic features, it is important that additional features, such as animacy, are not dependent on these features as well. Our linguistically uninformed model for animacy detection provides such an independent module.

The digitalization of large-scale cultural heritage collections such as the Dutch Folktale Database is often accompanied with traditional (text-based) search engines. We hope that our example of a semantic map of characters inspires researchers to disclose such collections in different and innovative ways.

Acknowledgments. The work on which this paper is based has been supported by the Computational Humanities Programme of the Royal Netherlands Academy of Arts and Sciences, under the auspices of the Tunes & Tales project. For further information, see `http://ehumanities.nl`.

References

1 Jelke Bloem and Gosse Bouma. Automatic animacy classification for Dutch. *Computational Linguistics in the Netherlands Journal*, 3:82–102, 2013.

2 Gosse Bouma, Gertjan Van Noord, and Robert Malouf. Alpino: Wide-coverage computational analysis of dutch. *Language and Computers*, 37(1):45–59, 2001.

3 Samuel Bowman and Harshit Chopra. Automatic animacy classification. In *Proceedings of the NAACL - HLT 2012 Student Research Workshop*, pages 7–10, 2012.

4 Bernard Comrie. *Language Universals and Linguistic Typology*. University of Chicago Press, 2nd edition, 1989.

5 Tom De Smedt and Walter Daelemans. Pattern for Python. *Journal of Machine Learning Research*, 13:2031–2035, 2012.

6 Daniel Dennett. *The Intentional Stance*. Cambridge, Massachusetts: The MIT Press, 1996.

7 Richard Evans and Constantin Orăsan. Improving anaphore resolution by identifying animate entities in texts. In *Proceedings of the Discourse Anaphora and Reference Resolution Conference*, pages 154–162, 2000.

8 Tao Gao, Brian Scholl, and Gregory McCarthy. Dissociating the detection of intentionality from animacy in the right posterior superior temporal sulcus. *The Journal of neuroscience: the official journal of the Society for Neuroscience*, 32(41):14276–14280, 2012.

9 Emiel Krahmer Jorrig Vogels and Alfons Maes. When a stone tries to climb up a slope: the interplay between lexical and perceptual animacy in referential choices. *Frontiers in Psychology*, 4(154):1–15, 2013.

10 Folgert Karsdorp, Peter Van Kranenburg, Theo Meder, and Antal Van den Bosch. Casting a spell: Indentification and ranking of actors in folktales. In F Mambrini, M Passarotti, and C Sporleder, editors, *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, pages 39–50, 2012.

11 Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4), 2013.

12 Theo Meder. From a dutch folktale database towards an international folktale database. *Fabula*, 51(1–2):6–22, 2010.

13 Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workship at ICLR*, 2013.

14 Joshua Moore, Christopher Burges, Erin Renshaw, and Wen tau Yih. Animacy detection with voting models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 55–60, 2013.

15 Mante S. Nieuwland and Jos J.A. van Berkum. When Peanuts Fall in Love: N400 Evidence for the Power of Discourse. *Journal of Cognitive Neuroscience*, 18(7):1098–1111, 2005.

16 John Opfer. Identifying living and sentient kinds from dynamic information: The case of goal-directed versus aimless autonomous movement in conceptual change. *Cognition*, 86(2):97–122, 2002.

17 Constantin Orăsan and Richard Evans. Learning to identify animate references. In Walter Daelemans and Rémi Zajac, editors, *Proceedings of CoNLL-2001*, pages 129–136, Toulouse, France, July, 6 – 7 2001.

18 Constantin Orăsan and Richard Evans. Np animacy identification for anaphora resolution. *Journal of Artificial Intelligence Research*, 29:79–103, 2007.

19 Lilja Øvrelid. Animacy classification based on morphosyntactic corpus frequencies: Some experiments with Norwegian nouns. In Kiril Simov, Dimitar Kazakov, and Petya Osenova, editors, *Proceedings of the Workshop on Exploring Syntactically Annotated Corpora*, pages 24–34, 2005.

20 Lilja Øvrelid. Towards robust animacy classification using morphosyntactic distributional features. In *Proceedings of the EACL 2006 Student Research Workshop*, pages 47–54, 2006.

21 Lilja Øvrelid. Linguistic features in data-driven dependency parsing. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2008)*, pages 25–32, 2008.

22 Lilja Øvrelid and Joakim Nivre. When word order and part-of-speech tags are not enough – Swedish dependency parsing with rich linguistic features. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 447–451, 2007.

23 Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

24 Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of The 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar, 2014.

25 Anette Rosenbach. Animacy and grammatical variation – findings from english genitive variation. *Lingua*, 118:151–171, 2008.

26 Roland Schäfer and Felix Bildhauer. Building large corpora from the web using a new efficient tool chain. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul, 2012. ELRA.

27 Jacques Sinninghe. *Volkssprookjes uit Nederland en Vlaanderen.* Kruseman, Den Haag, 1978.

28 Hans-Jörg Uther. *The Types of International Folktales: a Classification and Bibliography Based on the System of Antti Aarne and Stith Thompson*, volume 1-3 of *FF Communications.* Academia Scientarium Fennica, Helsinki, 2004.

29 Lauren Van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, pages 2579–2605, 2008.

30 Cornelis Van Rijsbergen. *Information Retrieval.* Butterworths, 1979.

The Love Equation: Computational Modeling of Romantic Relationships in French Classical Drama

Folgert Karsdorp[1], Mike Kestemont[2], Christof Schöch[3], and Antal van den Bosch[4]

1 **Meertens Institute**
Amsterdam, The Netherlands
`fbkarsdorp@fastmail.nl`

2 **University of Antwerp**
Antwerp, Belgium
`mike.kestemont@uantwerp.be`

3 **University of Würzburg**
Würzburg, Germany
`c.schoech@gmail.com`

4 **Radboud University**
Nijmegen, The Netherlands
`a.vandenbosch@let.ru.nl`

Abstract

We report on building a computational model of romantic relationships in a corpus of historical literary texts. We frame this task as a ranking problem in which, for a given character, we try to assign the highest rank to the character with whom (s)he is most likely to be romantically involved. As data we use a publicly available corpus of French 17th and 18th century plays (`http://www.theatre-classique.fr/`) which is well suited for this type of analysis because of the rich markup it provides (e.g. indications of characters speaking). We focus on distributional, so-called second-order features, which capture how speakers are contextually embedded in the texts. At a mean reciprocal rate (MRR) of 0.9 and MRR@1 of 0.81, our results are encouraging, suggesting that this approach might be successfully extended to other forms of social interactions in literature, such as antagonism or social power relations.

1998 ACM Subject Classification I.2.7 Natural Language Processing

Keywords and phrases French drama, social relations, neural network, representation learning

Digital Object Identifier 10.4230/OASIcs.CMN.2015.98

1 Introduction

Scholarship on literary texts has been among the seminal humanistic disciplines to engage with computational approaches [17], with e.g. Burrows's well-known study of Jane Austen's novels [6]. Burrows – and many others after him – have drawn attention to the potential of computational text analysis as a viable methodological complement to established, 'manual' approaches in literary criticism and narratological analysis. The social relations between Austen's characters, for instance, appeared to be reflected in their language use. In general, this kind of research has raised the question of the extent to which literary concepts can be formally modeled. In this paper, we focus on the linguistic aspects of romantic relationships in literary texts. We explore how this particular kind of social relationship can be modeled. We frame this research question as a 'matchmaking task': given a speaker, we try to assign

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 98–107
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the highest rank to the speaker with whom (s)he is most likely to be romantically involved on the basis of linguistic features.

The relationship between fictional characters in literary works can be viewed as a social network, the computational analysis of which has been steadily gaining popularity in recent years [15, 22]. When applied to literary fiction such as novels or plays, network analysis can yield insight into character relations in individual literary works or, more interestingly, reveal patterns and structure with regard to character networks in large collections of works. In this study, we analyze a collection of French plays from the 17th and 18th centuries. Relations between speakers are a central concern in research about dramatic works (see e.g. [19]), and love relationships are a type of speaker relation present in virtually any play from the period studied here. A basic assumption underlying our research is that love relationships in fiction are not only a matter of psychology, but are also a textual phenomenon which can be derived from the language used by speakers in a play. As a consequence, this study focuses on developing new methods for the formal modeling of love relationships in dramatic works based on speakers' linguistic behavior.

Among earlier work in this field is Moretti's essay 'Network Theory, Plot analysis' [14], in which the author draws on network theory to discuss the network of characters in Shakepeare's *Hamlet*, reminiscent of Knuth's classic network dataset [11] representing co-appearance patterns of characters in Victor Hugo's *Les MisÃ©rables*. A series of publications in the field of computational linguistics have further advanced a similar line of research in recent years, including social network analyses of e.g. nineteenth-century fiction [9]; *Alice in Wonderland* [1, 2], topic-model based approaches [7] and authorship attribution based on network features of novels [4]. A popularizing analysis of Marvel graphic novels has been presented in [3]. Few studies have explicitly focused on the formal modeling of love relationships in literary texts. Nevertheless, a number of inspiring studies have studied other sorts of specific social interactions e.g. friend-or-foe relationships [20] or antagonism ('good guy' vs 'bad guy') often in combination with methodologies from distributional semantics [5, 16].

This paper is structured as follows. We begin with a description of the French plays we used in Section 2. We then proceed with the methodology in Section 3 in which we discuss the task description, our evaluation method, the computational system and the features we used. Section 4 discusses the results of our study after which in Section 5 we conclude with some final remarks and starting points for further research.

2 The Data

The data for this study comes from the *Théatre classique* collection of French drama [10]. The collection contains 720 plays first published between 1610 and 1802, amounting to around 9.3 million word tokens. The plays vary in genre (with 340 comedies, 189 tragedies and 191 other sub-genres) and form (with 441 plays written in verse and 209 in prose only). The vast majority of plays have either one or five acts and 20–35 scenes. The plays are available as highly structured XML data encoded according to the guidelines of the Text Encoding Initiative (TEI P5) [8].[1] Each play's structure, in terms of acts and scenes, the cast members (henceforth, speakers) present in each scene, and their speeches, has been encoded in this markup. In addition, the XML files include detailed metadata about many of the roughly 6,500 speakers in the plays. In particular, the speakers' gender as well as their status with

[1] http://www.tei-c.org/release/doc/tei-p5-doc/en/html/DR.html

regard to love relationships have in many cases been explicitly encoded in the cast list, or can be inferred from the description of speakers in the cast list, as in the following example from Molière's *Le Dépit Amoureux*:

```
<castList>
  <castItem><role id="ERASTE" civil='M' type="H" statut='aristocrate' age='A'
             stat_amour='amoureux'>\'{E}RASTE</role>, amant de Lucile.</castItem>
  <castItem><role id="LUCILE" civil='F' type="H" statut='aristocrate' age='A'
             stat_amour='n\'{e}ant'>LUCILE</role>, fille d'Albert.</castItem>
[...]
</castList>
```

For the analyses presented here, we only used plays in which either such explicit annotation is available, or where it was possible to extract such information from the text provided in the cast list. Depending on the information available, we marked love relationships as either reciprocal or unidirectional. We extracted 295 love relationships from 200 different plays, of which only 90 could be assumed to be reciprocal. We created two datasets: one containing the 90 reciprocal relations, and one containing all 295 relationships, including all cases of unrequited love. We report results on both datasets.

3 Methods

Task Description. We cast our matchmaking problem as a ranking problem. Given a query speaker s_q from a particular play, the system should return a ranking of all other speakers in that play. The goal is to produce a ranking in which the highest rank is allocated to the true lover s_j. Framing our task as a ranking problem allows us to inspect the relation between a target speaker and the second-ranked speaker, who may be a contestant of the first-ranked speaker.

Learning to Rank. Learning to Rank is a supervised machine learning task which is to learn a ranking from observed data. Learning to Rank offers a simple, yet effective way to include heterogeneous features in one model. We make use of the sofia-ml toolkit [18] with the pegasos learning algorithm and the regularization parameter at its default value ($\lambda = 0.1$). As the algorithm randomly presents samples to the ranker, each run could produce slightly different results. All scores reported in this study are obtained by running the algorithm ten times with different random seeds, and taking the average over the results.

Evaluation. We test the performance of our system by means of leave-one-lover-out cross-validation. The training and test data are constructed in such a way that the query speaker s_q is only present in the test data and no relations to s_q are included in the training data. We evaluate our approach by means of the evaluation metric Mean Reciprocal Rank (MRR) [21] which computes the reciprocal of the rank at which the first relevant speaker (the true lover) was retrieved. MRR is a natural choice for our problem since in general, each speaker is at most in love with one other person. To evaluate the accuracy of the model we compute the MRR with a rank cutoff at 1.

3.1 Features

For each speaker in a play, we extract a vector containing the features described below. We scale each feature x within each query to the range $0 \leq x \leq 1$.

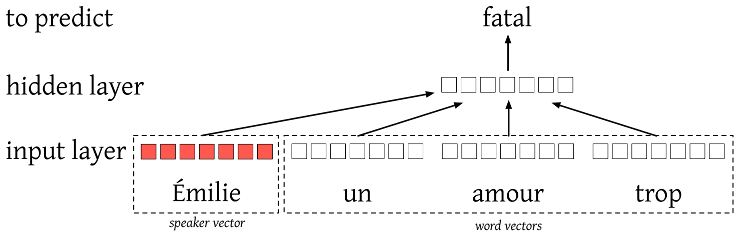


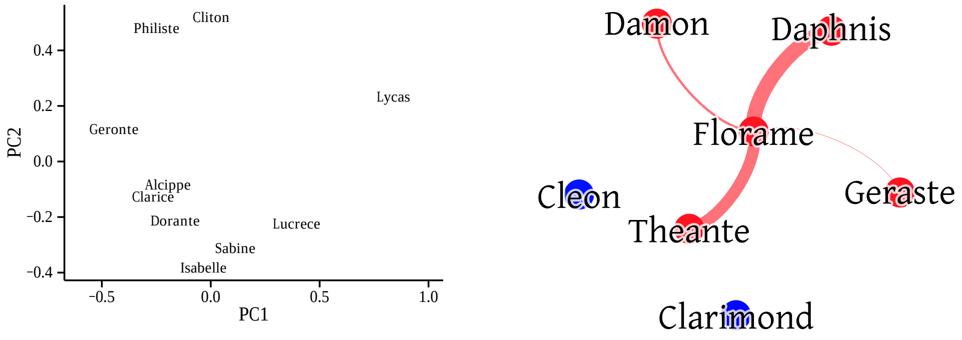
Figure 1 Graphical illustration of the Speaker Vector Model. In the model, each speaker has a unique vector representation (here milie in red). The speaker vector is activated each time a speaker generates an utterance.

3.1.1 Speaker Vectors

The first two features aim to capture information about the relationship between two speakers on the basis of their distributional semantics. For each speaker we want to learn a representation that aims to capture their semantic behavioral properties, such as the topics they speak of or the people they speak or think of. The approach we take to learn such representations is inspired by the recently proposed *Paragraph Vector* model [12]. This model is a shallow neural network that aims to learn dense, fixed-length semantic representations for arbitrarily long pieces of text. In the model, each paragraph (or any other chosen text unit, e.g. sentences or complete documents) is mapped to a unique vector of n dimensions. The words in the paragraphs are also mapped to a vector. However, these vectors are shared across word tokens, hence are not unique. The model initializes all vectors randomly. It then attempts to update the values along the dimensions by continuously predicting the next word in a particular context on the basis of these vectors. All vectors are trained using stochastic gradient descent. The dimensions (parameters) are updated by back-propagating the gradient through the network.

Our model learns dense representations not for individual paragraphs but for speakers. It does so in much the same way as the *Paragraph Vector* model, the only difference being that whereas the paragraphs in the original model are represented by a unique vector, a paragraph in our *Speaker Vector* model is mapped to the vector that belongs to the speaker of that paragraph. Figure 1 provides a graphical illustration of the model. The vector in red represents the vector of the speaker ï¿œÉmilie. Together with the context vectors for *un, amour* and *trop* the model attempts to predict the word *fatal*. The speaker vector of a speaker is activated during each utterance of that speaker and is used to predict each word in that utterance.

F1. Speaker Similarity For each candidate lover $s \in S$, where S is the set of candidate lovers in a play, we compute the cosine similarity between its vector representation and the vector representation of a query speaker s_q, $s_q \notin S$. The idea behind this feature is that we expect two lovers to speak of similar topics in similar ways, which should be reflected in their vector representations. To illustrate this point, in Figure 2a we present a two-dimensional reproduction of the speaker vectors in Pierre Corneille's comedy *Le Menteur* from 1644. The dimension reduction was generated through principal component analysis (PCA). The two lovers Alcippe and Clarice are placed adjacent to each other, reflecting the similarity of their vector representations. Interestingly, Alcippe's main contestant Dorante, the liar of the play's title, is close by. With some imagination, the plot visually expresses their contest around their object of desire, Clarice. To investigate



(a) 2-dimensional PCA projection of the speaker vectors in Pierre Corneille's *Le Menteur*. The two lovers Alcippe and Clarice are placed adjacent to each other, reflecting the similarity of their vector representations.

(b) Interaction network of Florame in Pierre Corneille's *La Suivante*. Florame predominantly interacts with two other speakers (depicted by the edge weight) of which Daphnis is his true love. Interestingly, Florame also often interacts with Theante who also is in love with Daphnis.

Figure 2 Visualization of features F1. and F5.

the overall effect of being a couple on the similarity between two speakers, we computed the pairwise cosine similarity between all lover and non-lover pairs within the same play. According to a two-sample Kolmogorov-Smirnov (KS) test, the two cosine similarity distributions differ significantly ($p < 0.0005$).

F2. Analogous Lovers The relation between Clarice and Alcippe can be described by their displacement vector D: $D(\text{Clarice}, \text{Alcippe}) = s_{\text{Clarice}} - s_{\text{Alcippe}}$, where s_{Clarice} is the vector representation of Clarice and Alcippe is represented by s_{Alcippe}. We can use this relation as a reference point to other possible relations between speakers. The similarity between a pair of displacement vectors, each describing a particular relation, should reflect the similarity between these relations. Given the relation between e.g. Clarice and Alcippe, we can compare other relations between speakers to this relation. Relations that are similar to that of Clarice and Alcippe are assumed to be romantic relationships. An illustrative example is the relation between Rosidor and Caliste from Pierre Corneille's highly complex early tragi-comedy *Clitandre*, first performed in 1630. Of all relations between Rosidor and any other speaker in the play, the one with Caliste is the one that is most similar to the relation between Clarice and Alcippe. We use this information in the following way. For each candidate lover $s \in S$ and a query speaker s_q, we compute the cosine similarity between the displacement vector $D(s, s_q)$ and the displacement vectors of all known lover couples. The maximum similarity between $D(s, s_q)$ and any other pair is used as the feature value. To assess the overall similarity between couples versus non-couples, we computed the maximum similarity between the displacement vectors of lover pairs to all other lover pairs and all non-lovers to all lover pairs. Again, the similarity distributions are significantly different (KS: $p < 0.0005$).

3.1.2 Word Vectors

Speaker vectors aim to capture topical properties of speakers. The similarity between two speaker vectors reflects the extent to which the two speakers speak of similar topics. Lovers also tend to speak about each other and often third parties talk about a couple. Speaker vectors do not necessarily capture this information, because most text in plays is in direct speech in which speakers refer to themselves by means of pronouns. To model the textual proximity of speakers we construct a version of the corpus in which each first person pronoun (*je, me, moi, mon, ma*) has been replaced by the unique ID of the speaker it refers to. Because speakers with the same name act in different plays, we also replace all proper names with the same unique ID. Essentially, this procedure is a cheap method to resolve co-references. We train word vectors on these adapted texts with 200 dimensions using the skip-gram and CBOW architecture [13].

F3. Word Similarity Similar to F1., for each candidate lover $s \in S$ we compute the cosine similarity between his/her word vector representation and the word vector representation of a query speaker s_q, $s_q \notin S$. On average, lovers have a cosine similarity of 0.58 while the mean cosine similarity between non-lovers is 0.34. As with the previous features, the similarity distributions are significantly different (KS: $p < 0.0005$).

F4. Word Analogy In a similar way as F2., we compute the maximum cosine similarity between the displacement vector $D(s, s_q)$ for candidate lover s and query speaker s_q and the displacement vectors of all known love couples. (KS: $p < 0.005$)

3.1.3 Physical Co-occurrence Features

The speaker vectors capture topical similarities and co-occurrence features present in the text. Not necessarily do these features reflect the physical co-occurrence of two speakers, for instance in a particular scene. The following two features aim to capture the physical co-occurrence of speakers. The idea behind these features is that two speakers are more likely to be in a love relationship if they meet more often.

F5. Interaction Frequency The first physical co-occurrence feature estimates the frequency of interaction between two speakers. Speaker s_i is in interaction with s_j if an utterance of s_i is preceded or followed by an utterance of s_j. For each speaker we compute the normalized count of how often (s)he interacts with another speaker. The result can be described as a network for each speaker in which weighted edges between two speakers are created if they interact. Edge weights are determined by the frequency with which the speakers interact. Figure 2b provides a graphical illustration of this feature in which we show the interaction network of Florame from Pierre Corneille's five-act comedy *La Suivante*, first performed in 1634. Florame predominantly interacts with two other speakers (depicted by the edge thickness) of which Daphnis is his lover. Interestingly, Florame also often interacts with Theante who is also in love with Daphnis. The overall interaction frequency distribution differences between couples and non-couples is significant (KS: $p < 0.0001$).

F6. Scene Co-occurrence The second physical co-occurrence feature is similar to F5. Here we construct a co-occurrence network for each speaker in a play in which edges between speakers are created if they appear in the same scene. The distribution differences between couples and non-couples are again significant (KS: $p < 0.0001$).

3.1.4 Meta Features

The XML-formatted versions of our plays provide rich metadata. One of the annotated features is the gender for each speaker. Given the dominance of heterosexual relationships in 17th and 18th century plays, we can apply an *a priori* filter on possible lover candidates on the basis of gender. To allow our system to be employed for different corpora that show

Table 1 Feature performance investigation. The first four columns provide the performance of the system with (individual) features on the full data set and the reciprocal data set. The last four columns show the performance of the system after removing the features mentioned.

feature	with feature				without feature			
	Reciprocal		Full		Reciprocal		Full	
	MRR	@1	MRR	@1	MRR	@1	MRR	@1
F1. Speaker Similarity	0.51	0.29	0.51	0.28	0.89	0.79	0.86	0.74
F2. Analogous Lovers	0.41	0.18	0.48	0.27	0.87	0.76	0.86	0.74
F3. Word Similarity	**0.74**	**0.59**	**0.73**	**0.56**	0.77	0.60	0.79	0.64
F4. Word Analogy	0.45	0.24	0.41	0.22	0.88	0.77	0.86	0.74
F5. Interaction Frequency	0.53	0.28	0.55	0.32	0.88	0.78	0.87	0.77
F6. Scene Co-occurrence	0.53	0.32	0.51	0.28	0.87	0.74	0.87	0.75
F7. Gender	0.29	0.07	0.37	0.12	**0.71**	**0.50**	**0.71**	**0.52**
F1. – F7.	0.9	0.81	0.87	0.75	–			

more variability in terms of the nature of relationships, we encode the gender of speakers as a feature.

F7. Gender For each combination of candidate lover $s \in S$ and the query speaker s_q, we compare their gender, where a gender difference is represented by a value 1 and gender identity by 0.

4 Results

Our Learning to Rank system shows promising results. The system achieves a Mean Reciprocal Rank of 0.9 on the dataset containing solely reciprocal love relationships and 0.87 on the full dataset. The MRR@1 (or accuracy) of the model on the reciprocal relationships is 0.81 and 0.75 on the full data set.

We performed an additional experiment in which for each feature we train our system using only that feature. The features in a Learning to Rank system can interact with each other in non-linear ways, implying that features that appear to have little effect in isolation may contribute strongly to the overall performance in combination with other features. We therefore also performed an ablation experiment in which for each feature we trained a system on the basis of all features except that feature. In Table 1 we present the results of the experiment that measures the performance of individual features (first four columns) and the results for the ablation experiment (last four columns).

In both the full data set and the data set containing solely reciprocal love relationships, the Word Similarity feature (F3.) is the best individually performing feature. The physical co-occurrence features (F4. and F5.) come next, followed by the Speaker Similarity feature (F1.) and the analogy-based features (F2. and F4.) The low performance of the gender feature is no surprise because it selects a number of speakers yet is unable to discriminate between them. In contrast, in the ablation experiment gender has the biggest contribution to the performance. Without the gender feature, the MRR drops from 0.9 to 0.71.[2]

[2] Note that this score is even lower than the score obtained by the Word Similarity alone. This suggests

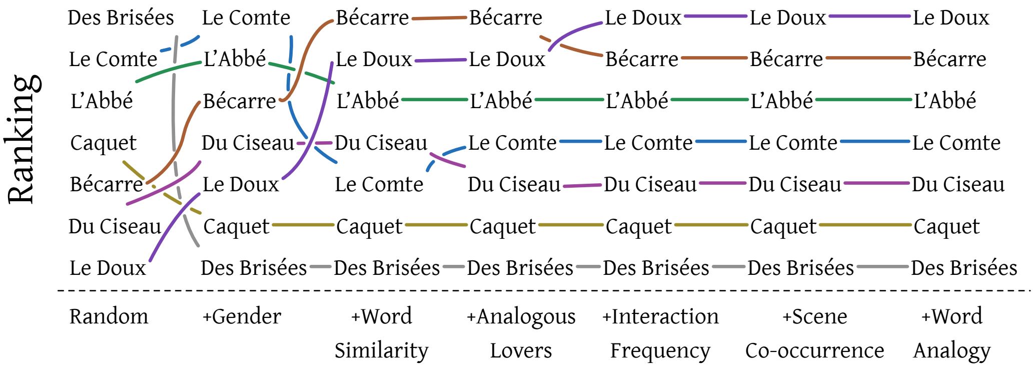


Figure 3 The effect of cumulatively adding features. We start with a random ranking (no features) and then continuously add more features. The plot visualizes how Le Doux, the lover of Suzanne in Beaunoir's *Le Sculpteur* gradually rises to the first position.

The gender feature acts as a sort of funnel that makes a pre-selection among possible love candidates. Given this pre-selection, the system makes a decision on the basis of the other features. To illustrate this process, we provide in Figure 3 the different rankings produced by the system for one speaker, Suzanne from Madame de Beaunoir's two-act prose comedy *Le Sculpteur* first performed in 1784. We start with a random ranking. The next ranking is based solely on the gender feature and puts all male speakers in the highest positions. As we add more features, Suzanne's lover Le Doux slowly rises to higher positions and takes over the first position from BÃ©carre when we add feature F5. Interaction Frequency.

5 Conclusions

The system for identifying romantic relationships in drama texts introduced here proves to be successful. We have shown that on the basis of textual and structural distributional properties of speakers in French drama texts we are able to confidently extract love relationships between speakers from the texts. These distributional properties function best in combination with knowledge about the gender of two speakers. Since knowledge about the gender of a potential couple is so important to our model and because we rely on manual annotations of this feature, the first point of future research should be the automatic classification of speaker gender. Next, we believe that our approach might be a fruitful starting point for modeling other relationships, such as well-know relations from structuralist analyses of drama, such as the triangle of protagonist, helper and antagonist [19].

One important limitation of the present setup is that the system can naively assume that all analyzed speakers are at least involved in one romantic relationship. The task is thus to identify, for a given speaker, the correct lover among a set of candidates. A more general, yet also more demanding task would be to predict for any given character, whether (s)he is romantically involved at all with another character. The distinction between both tasks is reminiscent of the difference between authorship attribution and authorship verification. With the former, resembling a police line-up, the system can assume that the correct author is present among the candidates. In the verification setup, however, the correct author is

that there are some interactions between features that actually harm the overall performance. We plan to investigate this in future work.

not necessarily included among the candidates. In future research, we hope to be able to generalize our model in this respect.

Our method could more generally serve as a heuristic tool for the exploration of large literary corpora and the serendipitous discovery of unsuspected speaker relations. Its ranking fosters investigations, for example, into what types of relations there are between the target speaker and the second-ranked speaker, who may for instance be a rival or a family member of the first-ranked speaker. More generally, our method is relevant in the context of increasing amounts of literary texts becoming available through large-scale digitization of our cultural heritage. Such textual data does not usually contain the rich annotations our data contains, and manually adding it is labor-intensive. Automatically extracting fundamental speaker relationships from raw text versions of plays helps gain a hermeneutically valuable access to such ever larger amounts of textual data.

Acknowledgments. The work of Folgert Karsdorp and Antal van den Bosch has been supported by the Computational Humanities Programme of the Royal Netherlands Academy of Arts and Sciences, under the auspices of the Tunes & Tales project. For further information, see `http://ehumanities.nl`. Mike Kestemont has been supported for this work as a postdoctoral researcher for the Research Foundation Flanders (FWO). Christof Schöch's contribution has been supported by funding from the German Federal Ministery of Education and Research under the eHumanities scheme (funding code 01UG1408); for more information, see `http://clgs.hypotheses.org/`

References

1 Apoorv Agarwal, Augusto Corvalan, Jacob Jensen, and Owen Rambow. Social network analysis of alice in wonderland. In *The proceedings of Workshop on Computational Linguistics for Literature, NAACL 2012*, pages 88–96, Montréal, Canada, 2012.

2 Apoorv Agarwal, Anup Kotalwar, and Owen Rambow. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013)*, pages 1202—-1208, Nagoya, Japan, 2013.

3 Ricardo Alberich, Joe Miro-Julia, and Francesc Rosselló. Marvel universe looks almost like a real social network. Preprint, arXiv id 0202174, 2002.

4 Mariona Coll Ardanuy and Caroline Sporleder. Structure-based clustering of novels. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 31–39, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

5 David Bamman, Brendan O'Connor, and Noah Smith. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 352—-361, Sofia, Bulgaria, 2013.

6 John Burrows. *Computation into criticism. A Study of Jane Austen's novels and an experiment in methods.* Clarendon Press, 1987.

7 Asli Celikyilmaz, Dilek Hakkani-Tur, Hua He, Greg Kondrak, and Denilson Barbosa. The actor-topic model for extracting social networks in literary narrative. In *NIPS Workshop: Machine Learning for Social Computing*, 2010.

8 TEI Consortium. *TEI P5: Guidelines for Electronic Text Encoding and Interchange.* TEI Consortium, 2014.

9 David K. Elson, Nicholas Dames, and Kathleen R. McKeown. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden, 2010.

10 Paul Fievre, editor. *Théâtre classique.* Université Paris-IV Sorbonne, 2007-2014.

11 Donald Ervin Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. ACM Press Series. ACM Press, 1993.

12 Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 2014.

13 Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.

14 Franco Moretti. Network theory, plot analysis. *New Left Review*, 68:80–102, 2011.

15 Mark Newman. *Networks. An Introduction.* Oxford University Press, New York, NY, USA, 2010.

16 Gabriel Recchia, Alexandra L. Slater, and Max M. Louwerse. Predicting the good guy and the bad guy: Attitudes are encoded in language statistics. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pages 1264–1269, 2014.

17 Susan Schreibman and Ray Siemens, editors. *A Companion to Digital Literary Studies*. Oxford: Blackwell, 2008.

18 D. Sculley. Large scale learning to rank. In *NIPS Workshop on Advances in Ranking*, pages 58–63, 2009.

19 Anne Ubersfeld, Frank Collins, Paul Perron, and Patrick Debbèche. *Reading Theatre*. Toronto Studies in Semiotics and Communication Series. University of Toronto Press, 1999.

20 Matje Van de Camp and Antal Van den Bosch. The socialist network. *Decision Support Systems*, 53(4):761–769, 2012.

21 Ellen M Voorhees and Dawn M Tice. The TREC-8 question answering track evaluation. In *Proceedings of the Eighth Text REtrieval Conference (TREC 8)*, volume 1999, page 82, 1999.

22 Scott Weingart. Demystifying networks, parts I & II. *Journal of Digital Humanities*, 1(1):9–21, 2012.

Learning Components of Computational Models from Texts*

Marjorie McShane[1], **Sergei Nirenburg**[1], **Bruce Jarrell**[2], **and George Fantry**[2]

1 **Rensselaer Polytechnic Institute**
Troy, NY, USA
{mcsham2, nirens}@rpi.edu

2 **University of Maryland, Baltimore**
Baltimore, MD, USA
bjarrell@umaryland.edu, Gfantry@medicine.umaryland.edu

Abstract

The mental models of experts can be encoded in computational cognitive models that can support the functioning of intelligent agents. This paper compares human mental models to computational cognitive models, and explores the extent to which the latter can be acquired automatically from published sources via automatic learning by reading. It suggests that although *model components* can be automatically learned, published sources lack sufficient information for the compilation of fully specified models that can support sophisticated agent capabilities, such as physiological simulation and reasoning. Such models require hypotheses and educated guessing about unattested phenomena, which can be provided only by humans and are best recorded using knowledge engineering strategies. This work merges past work on cognitive modeling, agent simulation, learning by reading, and narrative structure, and draws examples from the domain of clinical medicine.

1998 ACM Subject Classification I.6 Simulation and Modeling

Keywords and phrases cognitive modeling, simulation, clinical medicine, learning by reading

Digital Object Identifier 10.4230/OASIcs.CMN.2015.108

1 Introduction

New scientific findings are being published much faster than domain experts can read or developers of intelligent systems can integrate. One way to address this information onslaught is through automation: by configuring intelligent agents that engage in lifelong learning by reading. Ideally, such agents will initially be endowed with a cognitive model corresponding to the models held by domain experts; then, as the agents read new texts, they will compare the information reported in those texts to the current state of their cognitive model, incorporating time-stamped, source-stamped updates into the model. Agents thus modified will not only themselves show increasingly sophisticated behavior, they will be able to pass on this learning to both people and intelligent systems via updating applications. Although a human-quality realization of this vision is not achievable overnight, learning by reading *is* realistic and can be pursued in a way that offers benefits in the near-, mid- and long-terms.

* This research was supported in part by Grant N00014-09-1-1029 from the U.S. Office of Naval Research. Any opinions or findings expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research. Thanks to Mark Finlayson for useful conversations leading to our planned joint work.

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 108–123
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we explore the nature of computational cognitive models that are sufficient to support the physiological and cognitive simulation of human-like intelligent agents, as developed for a prototype virtual patient application. We describe how these models, like the human mental models that underlie them, are comprised of a data-attested sketch filled in by clinical reasoning and educated guessing. We show how automatic learning by reading has the potential to automate the acquisition and updating of the data-attested portions, but argue that the backbones of the models – which derive of largely unwritten human expertise – are still best crafted manually.

The clinical models of diseases to be discussed here have features both of scripts (in the Schankian sense [28]) and of narratives, which informs how we approach the task of learning by reading.

Like *scripts*, the models record typical sequences of events and the objects that participate in them. They also allow for extensive individualization of the dynamically simulated cases based on two factors: (1) the physiological, psychological, emotional and circumstantial features of each virtual patient instance, and (2) the "moves" of the virtual patient and the clinician with respect to diagnosis, treatment and patient lifestyle, which can be undertaken at any point in the patient's simulated life. While selecting individualizing features for each virtual patient leads to some aspects of determinism in the simulation, much of the simulation is open-ended because the moves of the live clinician interacting with the virtual patient are not known beforehand and can fundamentally change patient outcome.

Like *narratives*, clinical disease models involve a non-trivial – in fact, sometimes life-and-death – plot. Ideally, the patient and clinician cooperate to cure the patient, but conflict can also occur: e.g., the virtual patient can choose to lie to the doctor to cover up non-compliance with a treatment protocol, or it can refuse medical intervention due to its personality traits or phobias [14]. Although, from a developer's point of view, such behavior is expected (the virtual patient will have been endowed with personality traits giving rise to this behavior), from the point of view of a system user, such outcomes are expected to be viewed as unexpected plot elements.

At the junction of script and narrative are two additional features of our clinicial disease models. First, the models include attested but atypical – i.e., story-worthy – events. In fact, one of the motivating factors in developing this virtual-patient-oriented clinician training system was to expose medical trainees to the broadest possible set of disease manifestations during a temporally compact training experience. The second script-narrative bridge derives from the constant influx of newly reported medical knowledge that must be incorporated into the models. Such new findings, which are often reported in case studies, are similar to the unexpected plot twists of narratives which, once encountered, must be recorded as modifications to scripts.

Our goal of learning by reading involves the automatic detection of such new information, particularly from case studies, and its seamless incorporation into the core disease models. An enabling factor is the canonical plot-like structure of case studies, which provide summarized background knowledge supplemented by the plot twist of an unexpected patient experience.

The work reported here dovetails with several programs of research and development. Our focus on the medical domain reverberates with Sileno et al.'s [29] focus on the legal domain, and they, like us, seek to ultimately support automatic knowledge acquisition from narrative; however, whereas our work involves a formal knowledge base, language processing, and agent simulation, Sileno et al.'s contribution is at a more theoretical level. O'Neill and Riedl [27] and Finlayson [4] both present methods of generating narrative structures using a manually annotated corpus as input. Whereas O'Neill and Riedl do not commit to any particular knowledge representation formalism, Finlayson does, and uses it in the implementation of his Analogical Story Merging algorithm. Lieto and Damiano [6] discuss methods of detecting minimally different roles of participants in a narrative, such as hero vs. antihero. This aligns in spirit with our goal of detecting minimal differences between our disease models and the minimally different information presented in medical case studies. In terms of the ontologically-grounded modeling of complex events, the work of Schank and Abelson [28] was an early influence for the Theory of Ontological Semantics [21] that underpins the work reported here.

The paper is organized as follows. Section 2 sets the stage with an overview of the prototype medical teaching application – Maryland Virtual Patient (MVP) – that gave rise to our methodology of cognitive modeling. Section 3 draws a four-way comparison between human mental models, manually compiled cognitive models, the model components that can be semi-automatically elicited from human experts, and the model components that can be extracted from texts. Based on this comparison, we suggest a practical balance of effort between manual, semi-automatic and automatic knowledge acquisition strategies in support of agent configuration. Section 4 provides an overview of computational cognitive modeling in the OntoAgent environment, including excerpts from a disease model that successfully supported agent simulation in the MVP application. Section 5 describes how model components can be learned from texts, particularly by exploiting the predictable structure of genres such as case studies and disease overviews. Section 6 concludes the paper with the broader implications of this program of R&D.

2 The Maryland Virtual Patient (MVP) Application

Our modeling strategy developed during work on the prototype Maryland Virtual Patient (MVP) clinician training application [8] [9] [10] [13] [14] [22] [25] [26]. MVP is an agent-oriented system for automating certain facets of medical education and certification. It includes a network of human and software agents, at whose core is a virtual patient – a knowledge-based model of a person suffering from one or more diseases. The virtual patient is a "double agent" in that it displays both physiological and cognitive function. Physiologically, it undergoes both normal and pathological processes in response to internal and external stimuli, and shows realistic responses both to expected and to unexpected interventions; so if a trainee launches an inappropriate (unexpected) treatment, the patient's state will not improve and may even deteriorate, in which case the trainee must attempt to recover from his mistake.[1] Cognitively, the virtual patient experiences symptoms, has lifestyle preferences, can communicate with the human user in natural language, has memories of language interactions and simulated experiences, and can make decisions based on its knowledge of the world, its physical, mental and emotional states, and its current goals and plans. An optional tutoring agent provides advice and feedback to the trainee during the simulation.

Development of MVP follows the demand-side approach, meaning that it seeks to address a problem (detailed in [30]) that needs a solution rather than a problem that can be easily solved using standard methods (the supply-side approach). The specific problem MVP addresses is that medical educators, current training literature and pedagogical practice cannot provide medical students with adequately broad and varied training in cognitive analysis and problem solving. MVP seeks to permit trainees to diagnose and treat a large

[1] Compare this dynamic behavior with the static options in educationally-oriented branching scenarios that have also been called "virtual patients".

number of patient cases in a short amount of time, with the expectation that training results would mirror those of the SHERLOCK II electronic troubleshooting system for F16 aircraft of the US Air Force: participants using SHERLOCK II are reported to have learned more in 20 hours of tutoring than in 4 years of field experience [2].

Although many different paradigms of research and development involve entities called "virtual patients" (defined as mannekins, live actors, or branching scenarios), only MVP involves a knowledge environment that can support the approach to automatic lifelong learning described here. Key to this knowledge environment is reuse of the same knowledge representation language and static knowledge resources to support the wide range of agent functionalities described above [15]. Our prototype system has demonstrated that this AI-oriented, knowledge-based approach goes beyond theoretical status: we have worked out the details of knowledge representation and processing in implementations using realistic subject matter.

3 The Nature of Models

In this section we consider, in turn, human mental models, manually crafted computational cognitive models that seek to encode them, and the extent to which semi-automatic and automatic knowledge acquisition methods can realistically contribute to the computational modeling enterprise.[2]

Human mental models. Human mental models develop from a combination of experience, reading facts and stories, being told facts and stories, hypothesizing, reasoning, and even misrembering and forgetting. Although this wealth of contributors seems obvious, it is brought into relief when, as a non-specialist, one attempts to build a comprehensive computational model using only one of these sources as input: published texts. When working on modeling diseases and clinicial practices for MVP, the insufficiency of a "text-only" approach was immediately evident. Some gaps in knowledge represent facts that are actually not known because they are never measured: e.g., the physiological manifestations of the pre-clinical (non-symptomatic) stage of a disease. Other gaps reflect information that is not published in the literature for a *given* disease because it represents a broader generalization: e.g., a large tumor begins as a small tumor. Still other gaps reflect details that are not needed clinically (and are probably not known) but must be asserted if a realistic end-to-end simulation is to be implemented: e.g., does medication M, which ultimately cures disease D, improve property values at a steady rate or according to some non-linear function? The point is that humans somehow fill in these gaps sufficiently – albeit with a certain degree of uncertainty – to permit them to practice medicine effectively; and if they can do it, so must intelligent agents tasked with carrying out tasks requiring human-level reasoning.

Manually compiled, computational cognitive models. To develop computational cognitive models that were sufficient to support realistic patient simulations in MVP, a knowledge engineer led physican-informants through the process of distilling their extensive and tightly coupled physiological and clinical knowledge into the most relevant subset and expressing it in the most concrete terms. Not infrequently, specialists were also called upon to hypothesize about the unknowable, such as the preclinical stage of a disease and the values of physiological properties between the times when tests are run to measure them. Such hypotheses are

[2] See [5] for relevant discussion of manual vs. semi-automatic ontology development.

by nature somewhat vague, and could differ from expert to expert. However, rather than permit this imprecision to grind agent building to a halt, we proceed in the same way as live clinicians – and, presumably, any domain experts – do: by configuring *a* model that is reasonable and useful, with no claims that it is the only model possible or that it precisely replicates human functioning (cf. [1] for a discussion of modeling in the philosophy of science).

Decisions regarding what to include in our models derived from five desiderata: (1) that the models support realistic, interactive simulations; (2) that they not be unnecessarily detailed – i.e., if a detail would not be manifest in simulation (e.g., the firing of individual nerves), it was not included; (3) that they be easily updated to reflect new research findings; (4) that they be inspectable and explanatory, to support the pedagogical goals of the environment; and (5) that they be incorporated into an ontologically-grounded knowledge enviroment that supports all functionalities of all agents.

Taking these desiderata into account, and working within the OntoAgent cognitive architecture [15], we model diseases using an inventory of salient parameters whose values change over time in response to both internal stimuli (i.e., what the body does) and external stimuli (i.e., what the patient, doctor or outside world does). The selection of parameters to be included in a disease model is guided by practical considerations. Parameters are included because (a) they can be measured by tests, (b) they can be affected by medications or treatments, and/or (c) they are central to a physician's mental model of the disease. In addition to using parameters that directly reflect medically attestable properties, we also include abstract parameters that foster the formulation of a compact, comprehensible model (see Section 4 for examples).[3] Such features are particularly important at this stage of the discussion because they reflect the creative, unattested, aspect of computational modeling that naturally lies beyond automatic knowledge extraction methods since the information cannot be found explicitly in texts.

However, even if human reasoning is needed to build the more creative, hypothesis-driven aspects of computational models, the more concrete aspects can be aquired in semi-automatic and automatic ways, and it is to those that we now turn.

Semi-automatically acquirable model components. Since the collaboration between knowledge engineers and specialists is labor-intensive, the question arises, *To what extent can automation foster the process?* One way in which we experimented with reducing labor was by configuring a prototype knowledge elicitation system, called OntoElicit, to guide specialists through the process of independently recording "the basics" as preparation for work with a knowledge engineer [24]. The output of this work would then serve as input to the collaborative effort.

OntoElicit asks a domain expert to divide the given disease into conceptual stages correlating with important events. (The most obvious example of disease staging involves cancer, with its well-known stages 1 through 4; however, not all diseases are described in the literature as having a fixed inventory of stages.) Next, the system leads the expert through the process of providing – in a semi-formal way, guided by templates – details about disease progression, diagnosis and treatment. For example, when describing *physiology* and *symptoms*, the expert provides the inventory of properties that change over time, their start value before the disease begins and their expected values at end of each conceptual stage. Most values are recorded as a range of values covering different individual patients in the

[3] These features can be likened to the inclusion of intermediate categories in ontologies: although one does not typical talk about WHEELED-AIR-VECHILEs, this can be an appropriate node in an ontology.

population along with a default representing the most typical value. When describing *test results*, the expert indicates (a) which physiological properties are measured by each test, (b) any results that cannot be directly measured from the physiological model – e.g., visual findings by the administrator of the test, and (c) a "specialist's interpretation" of what the test results returned at that stage would indicate –e.g., "Suggestive of disease X." For *interventions* (medications, lifestyle changes, surgery, etc.), the expert indicates (a) which properties and/or symptoms are affected by the intervention, (b) the possible outcomes of the intervention, (c) possible side effects, and (d) if known, the percentage of the population expected to have each outcome and side effect. And for *diagnosis* and *treatment*, the expert provides fillers for ontological properties such as SUFFICIENT-GROUNDS-TO-SUSPECT (the given disease), SUFFICIENT-GROUNDS-TO-DIAGNOSE and SUFFICIENT-GROUNDS-TO-TREAT.

As mentioned earlier, the information acquired through OntoElicit is better described as *model components* than full models, since (a) some of the conceptual glue needed to hold the model together – most notably, causal chains – is absent and (b) the information is not written in the ontological metalanguage. However, the elicited information does include many aspects of a human mental model that would not be found in published sources, such as hypotheses about stage-by-stage disease progression despite the likely absence of actual attested property values for all stages. For this reason, the results of OntoElicit lie somewhere between a formal computational model and what we can expect to find in published sources.

Model components acquirable by agent reading. Published reports in the field of medicine typically contain only what is attested, making them insufficient as the sole source of knowledge for a comprehensive computational model. We might think of a complete computational model as a picture covered by a clear stencil whose holes represent model components that can be learned from the literature. As described in Section 5, the automatic learning of model components can be used either to update existing models or as the building blocks for more comprehensive, manually acquired models.

4 Modeling in OntoAgent

In the OntoAgent knowledge environment, disease models are recorded as complex events in the ontology. The ontology is a formal model of the world that is organized as a multiple-inheritance hierarchical collection of frames headed by concepts (OBJECTS and EVENTS) that are named using language-independent labels [7] [15] [21]. It currently contains approximately 9,000 concepts. The OBJECTS and EVENTS are described using PROPERTIES, both ATTRIBUTES and RELATIONS. The PROPERTIES themselves are primitives, i.e., their meaning is understood to be grounded in the real world without the need for further ontological decomposition. A short excerpt from the frame for the ontological concept SURGERY (which actually contains over a dozen more properties) is shown in Listing 1.

One of the properties not shown in this excerpt is the one that is key to modeling complex events: HAS-EVENT-AS-PART. The filler of this slot is an event script of the type introduced by Schank and Abelson [28]. Scripts represent typical sequences of events and their causal and temporal relationships. In other words, they encode how individual events hold well-defined places in routine, typical sequences of events that happen in the world, with a well-specified set of objects filling different roles throughout that sequence. Scripts require expressive means not provided in the simple slot-facet-filler formalism shown in Listing 1, and are recorded in a sister knowledge base. Scripts both drive agent simulation and support agent reasoning. For example, the script that describes a disease (its causes, variable paths of

Listing 1 Excerpt from the concept SURGERY in the OntoAgent ontology.

```
SURGERY
    IS-A (value MEDICAL-PROCEDURE)
    AGENT (default SURGEON) (sem PHYSICIAN) (relaxable-to HUMAN)
    THEME (default MEDICAL-PATIENT) (sem ANIMAL)
    LOCATION (default OPERATING-ROOM) (sem MEDICAL-BUILDING)
             (relaxable-to PLACE)
    INSTRUMENT (default SURGICAL-INSTRUMENT)
    DURATION (sem .5 - 8 (MEASURED-IN HOUR))
```

progression across patients, potential responses to interventions, etc.) permits (a) simulation of the disease in virtual patients, (b) reasoning about disease processes by the virtual medical tutor and (c) natural language dialog about the disease, since semantically-oriented natural language processing requires real-world knowledge support [21]. In short, a theoretically and practically motivated aspect of knowledge acquisition in OntoAgent is that knowledge, once recorded, should enable the maximum number of functionalities in the maximum number of agents [15].

For reasons of space, this discussion will focus primarily on the modeling of disease processes themselves, without as much detail about the modeling of interventions, clinical decision-making, agent decision-making, simulated agentive action, or any of the other necessary functionalities of agents, which are all handled in a corresponding way, as reported in the references cited earlier. It is important to understand the nature of the disease models in order to appreciate why they serve as a useful knowledge substrate for automatic knowledge acquisition from text. For this reason, we present select excerpts from our model for gastroesophageal reflux disease (GERD) by way of illustration.

4.1 An Excerpt from the Model for GERD

Gastroesophageal reflux disease, or GERD, can be defined as any symptomatic clinical condition that results from the reflux of stomach or duodenal contents into the esophagus. In laymen's terms, acidic stomach contents backwash from the stomach into the esophagus because the sphincter between the two – called the lower esophageal sphincter (LES) – is not functioning properly. The two sphincter abnormalities that give rise to GERD are abnormally low basal pressure of the LES (< 10 mmHg), or an abnormally large number or duration of so-called transient relaxations of the LES. Both of these lead to an increase in acid exposure to the lining of the esophagus. Clinically speaking, it does not matter which LES abnormality gives rise to excessive acid exposure, what matters is the amount of time per day this occurs. We record this feature as the variable "total time in acid reflux", or TTAR.

Although TTAR earns its place in the model as the variable that holds the results of the test called pH monitoring, it does not conveniently capture – for physicians or knowledge engineers – relative GERD severity. For that we introduced the abstract variable GERD-LEVEL. The values for GERD-LEVEL conveniently correlate with LES pressure as follows. If GERD is caused by a hypotensive LES, then GERD-LEVEL equals LES pressure. If GERD is caused by excessive transient relaxations, then the GERD-LEVEL reflects the same amount of acid exposure as would have been caused by the given LES pressure. So a GERD-LEVEL of 5 can indicate an LES pressure of 5 mmHg or a number/duration of transient relaxations per day that would expose the esophagus to that same amount of acid. Key aspects of the model then orient around GERD-LEVEL (rather than LES pressure, transient relaxations, or TTAR):

Table 1 Sample GERD levels and their associated total time in acid reflux (TTAR) per day. It also shows the baseline duration of each conceptual stage of the disease due to that TTAR, with more acid exposure leading to faster disease progression.

GERD level	ttar in hrs. per day	Stage duration in days
10	*less than 1.2*	*a non-disease state*
8	1.92	160
5	3.12	110
3	4.08	60

e.g., GERD-LEVEL is used to determine the pace of disease progression, with lower numbers reflecting more acid exposure and faster disease progression.

The stages of GERD are listed below. Each stage can be the end stage for some patients; that is, some lucky patients, even if left untreated, will never experience more than an inflamed esophagus, whereas others will end up with esophageal cancer. There is a bifurcation in disease path for patients experiencing late-stage disease, for reasons that are unknown.

- Preclinical: non-symptomatic inflammation of the esophagus.
- Inflammation: more severe inflammation of the esophagus, the beginning of symptoms.
- Erosion: one or more erosions occur in the esophageal lining.
- Ulcer: one or more erosions have progressed to the depth of an ulcer.
- *Post-ulcer path 1.* Barrett's metaplasia: a premalignant condition; progresses to cancer (an additional stage) in some patients.
- *Post-ulcer path 2.* Peptic stricture: an abnormal narrowing of the esophagus due to changes in tissue caused by chronic overexposure to gastric acid; does not lead to cancer.

The ontological scripts that support each stage of simulation include the basic physiological property changes, responses to interventions (if administered), and the effects of lifestyle choices. Sparing the reader the LISP code in which scripts are written, here is an example, in plain English, of how GERD progresses in an untreated patient who is predisposed to having erosion as the end stage of disease. During PRECLINICAL-GERD, the value of the property PRECLINICAL-IRRITATION-PERCENTAGE (an abstract property whose domain is MUCOSA-OF-ESOPHAGUS) increases from 0 to 100. When the value of PRECLINICAL-IRRITATION-PERCENTAGE reaches 100, the script for the PRECLINICAL-GERD is is unasserted, with the simultaneous assertion of the INFLAMMATION-STAGE script. During the INFLAMMATION-STAGE, the mucosal layer of the esophageal lining (recorded as the property MUCOSAL-DEPTH applied to the object ESOPHAGEAL-MUCOSA) is eroded, going from a depth of 1 mm. to 0 mm. over the duration of the stage. When MUCOSAL-DEPTH reaches 0 mm., the script for the INFLAMMATION-STAGE is unasserted, with the simultaneous assertion of the script for the EROSION-STAGE. At the start of the EROSION-STAGE, between 1 and 3 EROSION objects are created whose DEPTH increases from .0001 mm. upon instantiation to .5 mm. by the end of the stage, resulting in a decrease in SUBMUCOSAL-DEPTH from 3 mm. to 2.5 mm. When SUBMUCOSAL-DEPTH has reached 2.5 mm., the EROSION-STAGE script remains in a holding pattern since the patient we are describing does not have a predisposition to ulcer.

Over the course of each stage, property values are interpolated using a linear function, though other functions could be used if they were found to produce more lifelike simulations. So, halfway through PRECLINICAL-GERD, the patient's PRECLINICAL-IRRITATION-PERCENTAGE will be 50, and three quarters of the way through that stage it will be 75.

The length of each stage depends upon the patient's total time in acid reflux (cf. Table 1): e.g., a patient with a GERD-LEVEL of 8 will have a total time in acid reflux of 1.92 hours a day and each stage will last 160 days.

Some lifestyle habits, such as consuming caffeine, mints and fatty foods, increase GERD-LEVEL manifestation in some patients. In the model, if a patient is susceptible to GERD-influencing lifestyle habits and is engaging in those habits in simulation, then the effective GERD-LEVEL reduces by one. This results in an increase in acid exposure and a speeding up of each stage of the disease. If the patient is not actively engaging in the habit – e.g., after following the advice of a doctor to stop drinking caffeine – the GERD-LEVEL returns to its basic level. This is just one example of the utility of introducing the abstract property GERD-LEVEL into the model.

Let us now turn to two aspects of patient differentiation that highlight some more complex aspects of modeling: modeling why patients have different end stages of the disease, and modeling partial responses to medications. It is worth mentioning that we did not undertake either of these aspects of modeling in our initial model of GERD (published in [9]). The fact that we could seamlessly incorporate these enhancements, without perturbation to the base model, is evidence of the inherent extensibility of the models developed using this modeling strategy.

Modeling different end stages of disease across patients. It is unknown why patients have different end stages of GERD if the disease is left untreated. However, physicians can and do hypothesize about the reasons for cross-patient differentiation, which could include genetic, environmental, physiological and even emotional factors.[4] To capture some practically and pedagogically useful hypotheses, we introduced three abstract parameters into the model:

- MUCOSAL-RESISTANCE reflects the hypothesis that patients differ with respect to the degree to which the mucosal lining of the esophagus protects the esophageal tissue from acid exposure and fosters the healing of damaged tissue. A higher value on the abstract (0-1) scale of MUCOSAL-RESISTANCE is better for the patient.
- MODIFIED-TTAR combines MUCOSAL-RESISTANCE with the baseline TTAR to capture the hypothesis that a strong mucosal lining can functionally decrease the effect of acid exposure. For example, patients with an average MUCOSAL-RESISTANCE will have the stage durations shown in Table 1 above. Patients with an above-average MUCOSAL-RESISTANCE will have a lower MODIFIED-TTAR: e.g., if a patient's TTAR is 3.12 hours, but the patient has a mucosal resistance of 1.2, we model that as an MODIFIED-TTAR of 2.5 hours (3.12 multiplied by .8), and the disease progresses correspondingly slower. By contrast, if the patient's TTAR is 3.12 hours but it has a MUCOSAL-RESISTANCE of .8, then the MODIFIED-TTAR is 3.75 hours (3.12 multiplied by 1.2), and disease progression is correspondingly faster.
- DISEASE-ADVANCING-MODIFIED-TTAR is the total time in acid reflux required for the disease to manifest at the given stage. This variable permits us to indicate the end stage of a patient's disease in a more explanatory way that by simply asserting it. That is, for each patient, we assert how much acid exposure is necessary to make the disease progress into each stage, as shown in Table 2. If the acid exposure is not sufficient to support disease progression into a given stage (as shown by the italicized cells), the patient's

[4] For a medical description of the emotional effects on GERD, see [20]. For our incorporation of these factors into the clinical model, see [17].

Table 2 The first column indicates the patient's actual total time in acid reflux per day. The cells in the remaining columns indicate the total time in acid reflux needed for GERD to advance in that stage. Cells in italics show that the disease will not advance to this stage unless the patient's MODIFIED-TTAR changes – which could occur, e.g., if the patient took certain types of medications, changed its lifestyle habits or had certain kinds of surgery.

Patient	Modified-ttar	Preclin.	Inflamm.	Erosion	Ulcer	Pep.Strict.
John	1.92	1.92	1.92	*2.3*	*2.5*	*3.12*
Fred	2.8	1.92	1.92	2	2.7	*3.12*
Harry	4.08	1.92	1.92	3	3.5	4.0

Table 3 Effects of medications on MODIFIED-TTAR. The resulting MODIFIED-TTAR is written in brackets.

Patient	Modified-ttar	H2 blocker reduction	PPI once daily	PPI twice daily
John	1.92	.5 [1.42]	1.25 [.67]	1.5 [.42]
Fred	2.8	.3 [2.5]	1[1.8]	2.25 [.55]
Harry	4.08	.1 [3.98]	.8 [3.28]	2.2 [1.88]

disease will hit its end stage. For example, John is a patient whose disease will not progress past the Inflammation stage, even if left untreated, because his MODIFIED-TTAR is not high enough to support the erosion stage of GERD. Fred's disease will advance into the ulcer stage, and Harry's disease will advance to peptic stricture.

Modeling Complete and Partial Responses to Medication. In order to capture complete and partial responses to medications, medication effects are modeled as decreases in MODIFIED-TTAR, as shown in Table 3.

The table indicates the decrease in acid exposure caused by each medication for each patient, along with the resulting MODIFIED-TTAR. So, for each day that **John** takes an H2 blocker, his MODIFIED-TTAR will be 1.42, which is not a disease state. If he already has the disease, healing will occur. The other, stronger, medication regimens will also be effective for him. For **Fred**, the H2 blocker is not sufficient to promote complete healing (it brings the MODIFIED-TTAR down to 2.5), but it would be sufficient to not permit his disease to progress to the ulcer stage; or, if Fred were already in the ulcer stage, the ulcers would heal to the more benign level of erosions. If Fred took a PPI once or twice daily, his MODIFIED-TTAR would be < 1.92, meaning that his esophagus would heal completely. For **Harry**, the H2 blocker would not help at all – he would still progress right through the stricture stage. Taking a PPI once a day would heal ulcers and block late stages of disease. Taking a PPI twice a day would heal the disease completely, unless Harry had already experienced a stricture: there is no non-operative cure for a peptic stricture, a detail we will not pursue at length here but that is covered in the model (the STRICTURE object generated by the simulation remains a part of the patient's anatomy).

In sum, the physiologically-grounded parameter MUCOSAL-RESISTANCE permits each patient's end stage of disease progression to be calculated rather than asserted; and the parameters MODIFIED-TTAR and DISEASE-ADVANCING-MODIFIED-TTAR permit us to model full and partial efficacy of medications. As additional objective evidence becomes available through experimentation, the actual numerical values of these features can be modified accordingly.

Given models like this, the system need not exhaustively list all permutations of paths a trainee could take when diagnosing and treating a virtual patient, or all responses of the virtual patient to interventions. Instead, the system relies on these ontologically-grounded descriptions of basic physiology, disease processes, and effects of treatments and their interactions, so that the state of an MVP at any given time is dynamically computed by the system's reasoning module. Similarly, any of the tests available in the system can be run at any time, as they measure physiological properties of the patient as it lives its simulated life.

Let us conclude this section by returning to the question of how closely simulation-supporting computational models like these align with what is available in the published literature. The most striking difference is that much of our computational model is neither directly attested nor attestable, there being no widescale monitoring of people's physiology on a daily basis over the course of years. So, even those properties that are in principle measurable (such as TTAR and SUBMUCOSAL-DEPTH) are only a starting point for a picture that must be largely filled in by educated guesses. This is in addition to properties that are not currently measurable (such as PRECLINICAL-IRRITATION-PERCENTAGE) and properties that are introduced in order to capture specialists' generalizations about phenomena (e.g., GERD-LEVEL). The fact that clinicians' mental models are largely comprised of evidence-supported educated guesses does not impede effective clinical practice, but it does represent a divergence from the small subset of actually attested information in the literature. So, the question becomes, to what extent can we learn aspects of such models from texts?

5 Learning Model Components from Texts

The answer is that we can learn from texts *model components*, defined as ontologically-grounded property-value pairs that direcly contribute to full computational models. Learnable features have the following properties:

- They are straightforward and concrete, such as LES-PRESSURE (measurable by a test) or SENSITIVITY-TO-CAFFEINE (knowable based on patient reports); they are not abstract modeling properties (MODIFIED-TTAR, MUCOSAL-RESISTANCE), which will have no precise equivalents in published texts.
- They are known to be changeable over time, based on our ontological knowledge of the domain. For example, since we know that new medications and tests are constantly being invented, we know that the properties TREATED-BY-MEDICATION and ESTABLISHED-BY-TEST must have an open-ended inventory of values. By contrast, we do not expect the need to change the fact that heartburn can be a symptom of GERD, or that HEARTBURN-SEVERITY is modeled as having values on the abstract scale (0-1).
- (For knowledge involving causal chains only) If a sequence of events is modeled temporally rather than causally (using what we call "clinical knowledge bridges"), these can be automatically replaced by attested causal chains. However, if the model already records casual chains, their modification is likely to be too complex to be learned automatically without inadvertently perturbing the model.

Table 4 shows some examples of properties (associated with their respective concepts) whose values we believe can be learned from the literature.

The fillers for each property are formal, ontologically-grounded knowledge structures, which are produced during the automatic analysis of text by the OntoSem language processor. For example, all of the following text strings, and many more, will result in text meaning representations that permit the system to insert PROTON-PUMP-INHIBITOR as the value for the property HAS-TREATMENT of the concept GASTROESOPHAGEAL-REFLUX-DISEASE:

Table 4 Examples of properties, associated with their respective concepts, whose values can be learned from the literature.

Concept	Properties
DISEASE	HAS-EVENT-AS-PART, AFFECTS-BODY-PART, CAUSED-BY, HAS-SYMPTOMS, HAS-DIAGNOSTIC-TEST, HAS-TREATMENT
DIAGNOSTIC-TEST	MEASURES-PROPERTY, NORMAL-RESULT, ABNORMAL-RESULT, SIDE-EFFECTS, PAIN-INDUCED
MEDICAL-TREATMENT	HAS-EVENT-AS-PART, EFFICACY, HAS-RISKS, PAIN-INDUCED

- a proton pump inhibitor treats <can treat, can be used to treat, can be prescribed to treat, is often prescribed to treat> GERD
- GERD is <can be> treated by <cured by> (taking) a proton pump inhibitor
- doctors <your doctor may> recommend <prescribe> (taking) a proton pump inhibitor
- patients may <can, may be advised to> take a proton pump inhibitor

Establishing the functional equivalence of these strings is not done by listing; instead, it is done by combining our general approach to natural language understanding with algorithms for paraphrase detection ([11, 12]) and ontologically-grounded reasoning.

Let us consider just three examples of how natural language analysis supports the knowledge extraction process we are describing. Assume we are seeking to automatically learn or verify the veracity of the previously discussed fact "GASTROESOPHAGEAL-REFLUX-DISEASE (HAS-TREATMENT PROTON-PUMP-INHIBITOR)". As we said, all of the inputs above provide this information, albeit some more directly than others. The input *GERD is treated by a proton pump inhibitor* perfectly matches the lexical sense for the verb *treat* that is defined by the structure "DISEASE is treated by MEDICATION", and the analyzer generates exactly the text meaning representation we are seeking: GASTROESOPHAGEAL-REFLUX-DISEASE (HAS-TREATMENT PROTON-PUMP-INHIBITOR). In other cases, the basic text meaning representation includes additional "benign" information, which does not affect the truth value of the main proposition: e.g., the potential modality scoping over the proposition *GERD can be treated by a proton pump inhibitor* does not affect the truth value of the main proposition, which is the same as before and matches the expectation we seek to fill. In still other cases, the meaning we are looking for must be inferred from what is actually written. For example, the input *Your doctor may recommend a proton pump inhibitor* does not explicitly say that a proton pump inhibitor treats GERD, but it implies this based on the general ontological knowledge that a PRECONDITION for a physician advising a patient to take a medication is (DISEASE (HAS-TREATMENT MEDICATION)). Because the system has access to this ontological knowledge, it can make the needed inference and fill in our slot as before. It should be noted that these types of reasoning rules are not spontaneously generated – they must be recorded, like any other knowledge. However, once recorded, they can be used for any applicable reasoning need of the agent.

When investigating what information could be extracted from medical texts, we focused on two genres that offer different opportunities for knowledge extraction: case studies and disease overviews. Like narratives, both of these have largely predictable content and structure, which should support the automatic identification of disease model component information.

Case studies do not present all disease mechanics. Instead, they typically begin with a broad overview of the disease to serve as a reminder to readers who are expected to be familiar with "the script". Then they focus on a single new or unexpected aspect of the disease as manifest in one or a small number of patients (cf. the story-worthy aspects of

Table 5 Application for updating clinicians from case studies.

Case study: “Meditation as medication for GERD”
Author: Dr. J. Physician
Date: Jan. 11, 2018
Therapies for GERD
Mild: lifestyle modifications, H2 blocker, PPI QD, **MEDITATION-new**
Severe: PPI BID

narratives). For example, [3] is a case study that reports that a mother and daughter both suffer from the same rare disease, achalasia, and suggests that this case supports previous hypotheses of a genetic influence on disease occurrence. The new findings are typically repeated in the Abstract, Case Report, and Discussion sections, offering useful redundancy to improve system confidence.

The system can automatically compare the information in a case study with the ontologically grounded computational model as follows. First it can semantically analyze the case study, focusing on the TMR chunks representing the types of learnable property values listed above. (This focusing means that the system need not achieve a perfect analysis of every aspect of the text: it knows what it is looking for.) Then, it can compare the learned property values with the the values in the model. Continuing with our example of mother-daughter achalasia, our current model of achalasia has no filler for the value of CAUSED-BY since, when we developed the model, the cause was not definitively known (it still is not; the genetic influence remains to be validated). Automatically filling an empty slot with a new filler can be carried out directly, with no extensive reasoning necessary. However, the *nature* of that slot filler must be understood: it represents an instance, not a generic ontological fact. The system has two sources of evidence that this information is an instance: (1) the individuals spoken about are instances, so the features applied to them are also instances (compare this with assertions about about generic *people* or generic *you*); (2) the genre of case study sets up the expectation that reported information will be at the level of instance.

We believe it would be useful to configure an application that would alert clinicians to new findings in a “snaphot” formalism like that shown in Table 5. This presentation style encapsulates the expectations that: (a) clinicians know, without explanation, that one of the ontological properties of diseases is that they might have effective therapies; (b) when providing new information, it is useful to provide old information as the backdrop, with a clear indication of whether the new information adds to or overwrites the old information; (c) clinicians understand that information provided in case studies represents instances and not cross-the-boards generalizations; (d) modern-day users understand that entities can be clicked on for more information (e.g., which lifestyle modifications are being referred to), (e) terseness is appreciated by busy people operating within their realm of specialization.

Let us turn now to the other genre from which model information can be extracted: disease overviews. They typically present a stable inventory of properties of interest, often even introduced by subheadings, such as causes of the disease, risk factors, physiological manifestations, symptoms, applicable tests and procedures, and so on. Not surprisingly, these categories align well with the knowledge elements we seek to extract from texts, shown in Table 4. The natural language processing of disease overviews would proceed as described above. However, we envision applications for this processing to be somewhat different. For example, an application could respond to a clinician’s request for a thumbnail sketch of a disease by reading overviews, populating the inventory of key property values, and presenting them in a semi-formal manner, such a list of concept-property-value triples.

6 Discussion

This paper has presented a combination of work completed and work in the planning stages. The knowledge substrate and language processing capabilities are quite advanced, whereas the approach to mining new information from text is algorithmic.[5]

We present this work now as a contribution to a discussion that is key to computational narrative and agent building overall: to what extent can agents *in principle* learn models from text? And, if not full models, what *can* they learn through lifelong learning by reading?

In this paper we have suggested that although full models cannot be learned (they are largely unattested and rely centrally on educated guessing) certain model components can be automatically learned even in the near term, using currently available language processing technologies and achievable types of machine reasoning. This is a revolutionary idea, considering that we are talking about learning *ontologically-grounded knowledge structures* rather than extracting uninterpreted natural language strings from text.

If, by contrast, we want intelligent agents to learn full models from texts, then domain experts will need to write down fully specified mental models – an interesting prospect, particularly as it requires experts to boldly hypothesize about the unknown in the same way as they did to engineer the disease models for MVP. In short, modeling – be it recorded using an ontological metalanguage or a natural language like English – involves theorizing in an uncertain data space, something that is done as a matter of course in daily clinical practice but is not typically converted into published form. However, the potential rewards of fully specified (albeit with an understood tolerance for imprecision) models are tantelizing. Consider just a short excerpt from a committee report that lays out desiderata for virtual patient systems:

> "The clinician interacts with models and abstractions of the patient that place the raw data in context... These virtual patient models are the computational counterparts of the clinician's conceptual model of a patient... [The data] depict and simulate a theory about interactions going on in the patient and enable patient-specific parameterization... They build on submodels of biological and physiological systems..." [30].

Capabilities such as these directly motivate the need for inspectable, model-based artificial intelligence, not only in virtual patient applications but far beyond. It is our hope that the research reported here contributes to this vision, offering evidence of how component problems can be solved over time if we soberly analyze the necessary collaboration between human knowledge engineering and the potential for automatic agent learning.

References

1 Daniela Bailer-Jones. *Scientific Models in Philosophy of Science.* Pittsburgh, Pa: University of Pittsburgh Press, 2009.

2 Martha Evens and Joel Michael. *One-on-One Tutoring by Humans and Computers.* New Jersey and London: Lawrence Erlbaum and Associates, Publishers, 2006.

[5] We hope to further develop and implement the algorithms as a collaboration with Mark Finlayson, bringing to bear his Story Merging Algorithm [4], which will assist in comparing candidate model enhancements with our base models.

3 Yulia Viktorovna Evsyutina, Alexander Sergeevich Trukhmanov, and Vladimir Trophimovich Ivashkin. *Family case of achalasia cardia: Case report and review of literature*. World J Gastroenterol. Jan 28, 2014; 20(4): 1114–1118. Published online Jan 28, 2014.

4 Mark Finlayson. *Learning Narrative Structure from Annotated Folktales*. Doctoral Dissertation. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2012.

5 Antonio Lieto. *Manually vs Semiautomatic Domain Specific Ontology Building*. Thesis in Information and Commercial Electronics, University of Salerno, 2008.

6 Antonio Lieto and Rossana Damiano. *A hybrid representational proposal for narrative concepts: A case study on characters' roles*. In: Mark A. Finlayson, Jan Christoph Meister, and Emile Bruneau (Eds.), Proceedings of the 5th International Workshop on Computational Models of Narrative, 2014.

7 Marjorie McShane, Sergei Nirenburg and Stephen Beale. *An NLP lexicon as a largely language independent resource*. Machine Translation 19(2): 139–173, 2005.

8 Marjorie McShane, George Fantry, Stephen Beale, Sergei Nirenburg, and Bruce Jarrell. *Disease interaction in cognitive simulations for medical training*. In: Proceedings of MODSIM World Conference and Expo, Virginia Beach, Sept. 11–13, 2007.

9 Marjorie McShane, Sergei Nirenburg, Stephen Beale, Bruce Jarrell, and George Fantry. *Knowledge-based modeling and simulation of diseases with highly differentiated clinical manifestations*. In: R. Bellazzi, A. Abu-Hanna, J. Hunter (Eds.), Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME 07), Amsterdam, The Netherlands, 2007. Berlin, Heidelberg: Springer-Verlag. pp. 34–43. Amsterdam, The Netherlands, July 7–11, 2007.

10 Marjorie McShane, Bruce Jarrell, George Fantry, Sergei Nirenburg, Stephen Beale and Ben Johnson. *Revealing the conceptual substrate of biomedical cognitive models to the wider community*. In: J.D. Westwood, R.S. Haluck, H.M. Hoffman, G.T. Mogel, R. Phillips, et al. (Eds.), Medicine Meets Virtual Reality 16. Amsterdam, Netherlands: IOS Press, pp. 281–286, 2008.

11 Marjorie McShane, Sergei Nirenburg and Stephen Beale. *Resolving paraphrases to support modeling language perception in an intelligent agent*. In: J. Bos and R. Delmonte (Eds.), Semantics in Text Processing: STEP 2008 Conference Proceedings, Venice, Italy. London: College Publications, 2008.

12 Marjorie McShane, Sergei Nirenburg and Stephen Beale. *Two kinds of paraphrase in modeling embodied cognitive agents*. In: A.V. Samsonovich (Ed.), Biologically Inspired Cognitive Architectures. Papers from the AAAI Fall Symposium, Washington, DC, Nov. 7–9, 2008. AAAI Technical Report FS-08-04. Menlo Park, CA: AAAI Press, pp. 62–167, 2008.

13 Marjorie McShane and Sergei Nirenburg. *Dialog modeling within intelligent agent modeling*. Proceedings of the IJCAI-09 Workshop on Knowledge and Reasoning in Practical Dialog Systems, pp. 52–59. Pasadena, CA, USA, 2009.

14 Marjorie McShane, Stephen Beale, Sergei Nirenburg, Bruce Jarrell and George Fantry. *Inconsistency as diagnostic tool in a society of intelligent agents*. Artificial Intelligence in Medicine (AIIM), 55(3):137–48, 2012.

15 Marjorie McShane and Sergei Nirenburg. *A knowledge representation language for natural language processing, simulation and reasoning*. International Journal of Semantic Computing 6(1), 2012.

16 Marjorie McShane and Sergei Nirenburg. *Use of ontology, lexicon and fact repository for reference resolution in Ontological Semantics*. In: A. Oltramari, P. Vossen, L. Qin, and E. Hovy (Eds.), New Trends of Research in Ontologies and Lexical Resources, pp. 157–185, Springer, 2013.

17 Marjorie McShane, Sergei Nirenburg, Stephen Beale, Bruce Jarrell, George Fantry, and David Mallott. *Mind-, body- and emotion-reading*. Proceedings of IACAP 2013 (International Association for Computing and Philosophy), University of Maryland College Park, July 15–17, 2013.

18 Marjorie McShane, Sergei Nirenburg and Bruce Jarrell. *Modeling decision-making biases*. Biologically-Inspired Cognitive Architectures (BICA) Journal. Volume 3:39–50, 2013.

19 Nighat F. Mehdi, Miles M. Weinberger and Mutasim N. Abu-Hasan. *Achalasia: unusual cause of chronic cough in children*. Cough 2008, 4:6.

20 I. Mizyed, S.S. Fass and R. Fass. *Review article: gastro-oesophageal reflux disease and psychological comorbidity*. Alimentary Pharmacology and Therapeutics, 29: 351–358. Blackwell Publishing, Ltd., 2009.

21 Sergei Nirenburg and Victor Raskin. *Ontological Semantics*. The MIT Press, 2004.

22 Sergei Nirenburg, Marjorie McShane and Stephen Beale. *A simulated physiological/cognitive "double agent"*. Proceedings of the Workshop on Naturally Inspired Cognitive Architectures, AAAI 2008 Fall Symposium, Washington, D.C., Nov. 7–9, 2008.

23 Sergei Nirenburg, Marjorie McShane and Stephen Beale. *Aspects of metacognitive self-awareness in Maryland Virtual Patient*. Proceedings of the AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems, Nov. 11–13, Arlington, VA, 2010.

24 Sergei Nirenburg, Marjorie McShane and Stephen Beale. *Hybrid methods of knowledge elicitation within a unified representational knowledge scheme*. In: J. Filipe and J.L.G. Dietz (Eds.), KEOD 2010: Proceedings of the International Conference on Knowledge Engineering and Ontology Development, October 25-28,Valencia, Spain, 2010. SciTePress. pp. 177–182, 2010.

25 Sergei Nirenburg, Marjorie McShane, Stephen Beale, Jesse English and Roberta Catizone. *Four kinds of learning in one agent-oriented environment*. In: A.V. Samsonovich et al. (Eds.), Proceedings of the First International Conference on Biologically Inspired Cognitive Architectures, Arlington, VA, Nov. 13-14, 2010. Amsterdam, Netherlands: IOS Press, pp. 92–97, 2010.

26 Sergei Nirenburg, Marjorie McShane, Stephen Beale, Bruce Jarrell and George Fantry. *Intelligent agents in support of clinical medicine*. Proceedings of MMVR18, February 9–12, 2011, Newport Beach, California, 2011.

27 Brian O'Neill and Mark Riedl. *Applying qualitative research methods to narrative knowledge engineering*. In: Mark A. Finlayson, Jan Christoph Meister, and Emile G. Bruneau (Eds.), Proceedings of the 5th Workshop on Computational Models of Narrative, pp. 139–153, 2014.

28 Roger Schank and Robert Abelson. 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. New York: L. Erlbaum Associates.

29 Giovanni Sileno, Alexander Boer, and Tom van Engers. *Legal Knowledge Conveyed by Narratives: Towards a Representational Model*. In: Mark A. Finlayson, Jan Christoph Meister, and Emile G. Bruneau (Eds.), Proceedings of the 5th Workshop on Computational Models of Narrative, pp. 182–191, 2014.

30 W. W. Stead and H. S. Lin (Eds). *Computational Technology for Effective Health Care: Immediate steps and strategic directions*. National Research Council. Washington, DC: The National Academies Press, 2009.

31 C. B. Thielst. *The new frontier of electronic, personal and virtual health records*. Journal of Healthcare Management, vol. 52, pp. 75–78, 2007.

Cross-Document Narrative Frame Alignment

Ben Miller, Ayush Shrestha, Jennifer Olive, and Shakthidhar Gopavaram

Georgia State University
Atlanta, GA, U.S.A
miller@gsu.edu, jolive1@gsu.edu, sgopavaram1@cs.gsu.edu, ayush.shrestha@gmail.com

Abstract

Automated cross-document comparison of narrative facilitates co-reference and event similarity identification in the retellings of stories from different perspectives. With attention to these outcomes, we introduce a method for the unsupervised generation and comparison of graph representations of narrative texts. Composed of the entity-entity relations that appear in the events of a narrative, these graphs are represented by adjacency matrices populated with text extracted using various natural language processing tools. Graph similarity analysis techniques are then used to measure the similarity of events and the similarity of character function between stories. Designed as an automated process, our first application of this method is against a test corpus of 10 variations of the Aarne-Thompson type 333 story, "Little Red Riding Hood." Preliminary experiments correctly co-referenced differently named entities from story variations and indicated the relative similarity of events in different iterations of the tale despite their order differences. Though promising, this work in progress also indicated some incorrect correlations between dissimilar entities.

1998 ACM Subject Classification I.2.7 Natural Language Processing

Keywords and phrases computational narrative, natural language processing, graph theory, text mining

Digital Object Identifier 10.4230/OASIcs.CMN.2015.124

1 Introduction

Building parse tree representations of sentence-level grammars and comparing those representations to assess grammatical similarity has been an achieved goal of natural language processing (NLP), at least in English, since the development of the Penn Treebank and the success of statistical parsers in the mid-1990s [19]. Adapting this kind of parse tree comparison approach to higher-level analyses such as cross-document comparison of narrative similarity, however, remains an open challenge. The goal of this preliminary research is to advance our prior work in narrative information extraction [22] and visualization [28] for narrative similarity assessment, event alignment, and cross-document coreference using a graph comparison approach. Our method uses matrix representations of the graphs where each node is an entity, each edge is a relation, and each matrix represents one "event" as denoted by the language processing tool EVITA [26]. For this study, an entity is either a character, a location, or an organization.

Humanities scholars focus on broad problematics such as semantics, representation, narrative: problematics that frequently bridge, fracture, and co-referentially scatter throughout documents and corpora. Discourse analysis [14] and TextTiling [13] are two methods used to circumvent sentential boundaries by segmenting documents into blocks according to inferred characteristics of speaker, function, or character frequency change boundaries. As with topic

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 124–132
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

modeling methods like latent semantic analysis [8], these blocks facilitate comparisons of macro-level structures. These segmentation methods might produce blocks roughly equivalent to scenes. However, they rely on string and semantic vectors and have no particular sensitivity to features key for the structural analysis of narrative. Our research instead expands on graph comparison methods, which can more readily be made sensitive to narratological features such as events. Comparison of narrative graphs facilitates 1) alignment of event descriptions across narratives, 2) cross-document co-reference, and 3) the testing of structuralist narratological schema. To preliminarily test one and two, we implemented a method as described below.

Structural analyses of narrative successfully identified elements significant for the composition and study of narrative. Russian formalists such as Propp [25] and later work by Genette [11], Bal [1], and others yielded many complementary top-down models for deconstructing narratives. These schema generally distinguish between fabula and discourse: events to be narrated and the nature of that narration, respectively. Discourse order is the relationship between the temporality of events and their representation as part of a narrative [11]. This structural perspective serves humanists well when analyzing single narratives or small corpora but is highly subject to interpretation, and therefore operationalizes poorly. Computational models developed from formalist approaches have been the subject of compelling experiments. Like work by Finlayson on analogical story merging [9] and Fisseni on story comparison [10], our work presents a bottom-up method reliant on top-down narratological schema. Unlike theirs, our work focuses on unsupervised cross-document comparison of events and characters.

This method facilitates cross-document narrative analysis by indicating the similarity of a character's relationships across different tellings of a particular story and by allowing for the comparison of event language. Although much work remains and the anaphora resolution task was manually verified, this method would work with larger corpora as extraction, lookup, and comparison operate in an unsupervised manner.

2 Method

Comparison of events across documents relies on the production of structured representations of events. In the case of this study, that structure is a matrix of entity-entity relations for each event. Generalizing the specific language of a story is necessary as abstracted language facilitates comparison. This study used event hypernym sequences to generalize from the specific language of a given event. After identifying language features that are indicative of events, identifying the entities present in that event, and finding the hypernym of the lexical feature identified as the verb or state of the event, matrices were produced. Some language features indicative of events include finite clauses, event-referring nouns, and nominalized noun phrases [26]. Comparison via a neighborhood similarity function provided our primary comparison method to highlight event and character similarities.

2.1 Extraction

Events were automatically marked in the narratives using the Events in Text Analyzer (EVITA). EVITA uses statistical and linguistic approaches to identify and classify the language denoting orderable dynamic and stative situations [18]. EVITA's overall accuracy in event recognition was found by [18] to be $80.12\%F\_\{\beta\} = 1$ over TimeBank, with 74.03% precision and 87.31% recall. [18] summarizes evaluations of related work in automatic event detection, including TimeML [5], STEP [3], and event recognition using a multiclass classifier [20]. Their summary findings showed that EVITA either outperformed or was competitive

Table 1 Adjacency matrix created from one version of “Little Red Riding Hood". An edge (in the graph) or 1 (in the adjacency matrix) between two entities signify that these entities interacted within the given set of events.

	lrrh	wolf	grandmother	woodcutters	forest	gm_house
lrrh	1	0	0	1	1	0
wolf	0	1	0	0	1	0
grandmother	0	0	1	0	0	0
woodcutter	1	0	0	0	1	0
forest	1	1	0	1	0	0
gm_house	0	0	0	0	0	1

with other automated solutions. A more robust theoretical model for what constitutes an event is being developed for implementation by the NewsReader project in [31].

EVITA sequentially numbers events. That sequence must stand in for discourse order because fiction frequently lacks the dates and timestamps necessary to identify story order. They features are also necessary for discrete temporal language taggers like SUTime [7] and GUTime [32]. Entity extraction and anaphora resolution was accomplished using the Stanford Named Entity Recognizer (NER) followed by manual verification; entity classification was not relevant for this method as all three types of NE were identically represented in the matrices.

2.2 Graph Creation

Given an extracted set of events from a document, E_1 to E_n, we first divide them into k subsets ordered according to the story time. Event subsets can be defined in various ways: by manual adudication according to various criteria, or automatically by document section, by prevalent entities, by location shifts, or by prevalent event types. For this experiment, we ran the process two with manually defined event subsets based on location shifts, and with no event subsets.The number of events is determined by the event analyzer. The number of subsets is variable but currently must match from story to story. All entities (characters and locations) associated with the events are listed on a per-event basis. Each version of the story included a subset of some version of Little Red Riding Hood, mother, home, wolf, grandmother, woodcutters, forest, and grandmother’s house as key entities.

Following this process, we create a graph with these entities for every event subset. We begin by treating each entity as a vertex and adding an edge between verticies if both are present in the same event within an event subset. An adjacency matrix representation of a subset is shown in Table 1. In this subset of events, Little Red Riding Hood and the woodcutters are present in the forest in a particular event (the value is 1). In the same subset, the wolf is also in the forest. However, the wolf does not meet Little Red Riding Hood in any of the events in this subset, thereby resulting in no edge between them (the value is 0).

2.3 Similarity Analysis

Many domain-specific algorithms to compute similarity have been developed. Most are based on neighborhood analysis. Considering the problem of narrative frame alignment in this context treats a narrative as a directed graph; each event leads to the next and each set of events constitutes a group or neighborhood. That perspective allows for event or story analogy to be considered using the more robust methods applied to network similarity problems. In this paper, we propose our own similarity analysis method inspired by the work of Blondel et al [4].

Given a document, A, let p be the total number of entities in the document. If the set of events in this document are divided into k parts, we can represent the events in the document as a $3D$ matrix: $A_{p,p,k}$. The number of parts is some number equal to or less than the total number of event segments. Let $B_{q,q,r}$ be another document with q entities and r parts. Likewise, the number of parts is some number equal to or less than the number of events in that story. We compare each adjacency matrix in A with the corresponding adjacency matrix in B. In cases where $k \neq r$, we reduce to zero and pad the smaller matrix to the bigger size. For each adjacency matrix, as in the hyperlink-induced topic seach (HITS) inspired algorithm [15] proposed by [16], we compute

$$X \leftarrow BXA^T + B^TXA \tag{1}$$

and normalize X after each iteration. HITS was developed to facilitate search on the web by assessing the authority and role of nodes in large graphs; [16] extended that algorithm to the problem of identifying topological similarities in large, sparse, isomorphic graphs. That structure corresponds to the graphs that result from our event and entity extraction processes. The even iterations converge to a final similarity matrix. To simplify and speed up this process, we use the Kronecker product and the vec(.) operator. This process results in

$$x \leftarrow (A \bigotimes B + A^T \bigotimes B^T)x \tag{2}$$

where $x = vec(X)$. This set of equations give a similarity score frame per scene (part), which is then aggregated to produce a final similarity score between the stories.

3 Preliminary Experiment

For the purposes of testing our methodology, we selected 10 of the 58 known iterations [29] of the Aarne-Thompson type 333 story (ATU333), “Little Red Riding Hood.” Those 10 iterations are from [12, 33, 27, 21, 24, 2, 30, 6]. This corpus of 10 was compiled and selected to represent the canonical versions of the ATU333 story and significant variations from that story (e.g., where the wolf was the hero). The purpose of compiling and using this corpus was to begin our testing with a story featuring a high degree of narrative overlap. That overlap let us test the method on fine-grain distinctions between re-tellings. While our method benefits from such homogeneous narrative content, we believe that analyses of other narrative corpora with overlapping sets of events would be equally viable because of the highly granular event segmentation, the hypernym language abstraction procedure, and the binning of entity classifications into a single entity category.

$1,384$ events were extracted via this method across 10 story versions. Numbering 8,450 tokens, including titles and authorship information, the overall density of extracted events to tokens is high. Contrasted to event detection methods reliant on temporal expressions such as SUTime, which only identified two events in the corpus, this density of event detection provides a good basis on which to compare narrative structure. Generalizing event keywords from specific tokens to hypernyms of those tokens (e.g., event 41 from [6]: “armed” lemmatized to “arm” of which the hypernym found via WordNet [23] is “supply”) preserves the function of each event within the story but allows for storytelling variation. The current method for finding the hypernym looks for agreement across all results returned by WordNet. In the case of disagreement, the hypernym most frequently returned is selected; in the case of a tie, the first hypernym is used. The automatically produced matrices for this work are exemplified by Table 2. The stack corresponds to the “Oh, grandmother, what big ears you have!” to “[a]nd with that he jumped out of bed, jumped on top of poor Little Red Cap, and ate her up” sequence from [17].

Table 2 shows six layers from the $3D$ event matrix stack. The current language processing pipeline finds the events hypernyms but does not use them to assess narrative similarity. Results of functions (1) and (2) on the adjacency matrices are exemplified below in Table 3. Column headings correspond to entities from [12] for event 3, and row headers correspond to entities from [17] for event 4.

Table 3 shows that the measure of similarity between Little Red Riding Hood (“lrrh”) and Little Red Cap (“lrc”) is 0.32. Although low, that score was calculated only based on entity-entity connections and the sequence of those connections. When examined on the basis of an individual event, of which [17] contains 122, the correlations are unremarkable. Effectively, the wolf could be seen as similar to Rotkäppchen as to the woods. It is only when aggregates of events are compared that the method begins to correctly indicate entity similarities across documents.

Table 4 shows the potential for this method to align characters from different versions based upon their position within the story. It presents the similarity comparison for all events across two iterations of the story, summing all event matrices for two variations. Version 1 occupies the columns (Little Red Riding Hood, Wolf, Grandmother, Woodcutters, Home, Forest, and Old Woman’s House) and version 2 the rows (Little Red Cap, Wolf, Grandmother, Huntsman, Home, Woods, Grandmother’s House). Name independent character similarity is demonstrated by the 0.94 correspondence between the two wolves.

The event matrix suggests that certain characters function dissimilarly between variations: most notably, Grandmother. The corresponding value between the Grandmother characters is only 0.31, suggesting that they share some event associations but not as many as are held by other cross-document pairings. That assessment is accurate as, in version 1, the story concludes upon the wolf’s consumption of both Little Red Riding Hood and Grandmother. In version 2, both survive to boil a second hungry wolf. Table 5 compares version 2 and version 6, a more modern iteration, showing promising albeit imperfect results.

In Table 5, we see the method correctly correlate two principal characters in the story, a process we refer to as alignment. It also suggests strong correlations between each of those two characters and their respective wolves. However, for many of the other principal characters, it is not the highest similarity score that suggests correct character alignment, but rather the second highest similarity. The wolf in version 6 is seen as 0.86 similar to Rotkäppchen but only 0.62 similar to the wolf from version 2. Other less well-documented characters simply do not seem to show up frequently enough to be susceptible to alignment. One takeaway from this preliminary work is that it may only be a viable method for characters that frequently appear in stories. Another compelling way to read this table, however, is to compare the similarity of two characters from two different works against each other. For example, version 6’s Little Golden Hat is seen as more similar to both the wolf and the woods than her counterpart, Rotkäppchen. That way of reading the results of our method suggests that we can both identify which characters are most similar between two versions of a story and compare the varying similarity of a character between versions of a story.

4 Conclusion and further work

This preliminary work resulted in a viable method for narrative alignment and for the cross-document coreference of characters bearing different names but similar story functions.

Table 2 Six matrix layers from 3*d* stack of event matrices.

Event		LRRH	Grandmother	Wolf
106 – undergo	Bed	1	1	1
107 – perceive	Bed	1	1	1
108 – undergo	Bed	1	1	1
109 – seize	Bed	1	1	1
110 – undergo	Bed	1	1	1
111 – consume	Bed	1	1	1

Table 3 Character similarity across “Little Red Riding Hood” and “Rotkäppchen”.

	LRRH	Wolf	Grandmother	Woodcutters	Home	Woods	OWH
LRC	.32	.25	0	.25	0	.32	0
Wolf	.32	.25	0	.25	0	.32	0
Grandmother	0	0	0	0	0	0	0
Huntsman	0	0	0	0	0	0	0
Home	0	0	0	0	0	0	0
Forest	.32	.25	0	.25	0	.32	0
Grandmother’s	0	0	0	0	0	0	0

Table 4 Character similarity across all events for “Little Red Riding Hood” and “Rotkäppchen”.

	LRRH	Wolf	Grandmother	Woodcutters	Home	Forest	OWH
LRC	**.67**	.76	.31	.14	.14	.48	.37
Wolf	.79	**.94**	.42	.14	.14	.56	.5
Grandmother	.35	.47	**.31**	0	0	.16	.37
Huntsman	.23	.28	.18	**0**	0	0	.26
Home	0	0	0	0	**0**	0	0
Woods	.48	.53	.16	.14	.14	**.48**	.16
Grandmother’s	.39	.52	.34	0	0	.16	**.42**

Table 5 Character similarity across all events for “Little Golden Hat" and “Rotkäppchen".

	LGH	Mother	Grandmother	Wolf	Wood	Grandmother’s	Woodcutters
LRC	**1.00**	0.06	0.45	0.86	0.06	0.24	0.10
Mother	0.04	**0.01**	0.07	0.03	0.00	0.03	0.00
Grandmother	0.61	0.09	**0.32**	0.55	0.07	0.12	0.01
Wolf	0.79	0.05	0.21	**0.62**	0.05	0.23	0.01
Woods	0.21	0.03	0.06	0.13	**0.04**	0.05	0.01
Grandmother’s	0.05	0.00	0.12	0.04	0.01	**0.04**	0.00
Huntsman	0.10	0.00	0.00	0.09	0.00	0.00	**0.00**

Story function is being used here principally to describe the social function of a character or location relative to other characters and locations. It was determined by segmenting the story into a series of events, then identifying character-character and character-location relations and the order of those relations. The event segmentation, relation extraction, and matrix comparison methods are implemented and tested. The hypernym extension of our method will divide the event hypernyms into overlapping three-window sequences of two-to-four terms each corresponding to past, present, and future states. Those sequences will be used as weighting functions on the Kronecker product for the cross-document comparison of narrative frame similarity. For example, the entity relationships in the matrix representing a sequence of three events in document A and the entity relationships in the matrix representing a sequence of three events in document B will be factored against each other with the relative similarity multiplied by the similarity score of the hypernym sequence. Three identical terms in each window frame of past, present, and future will score as a 1. No common hypernyms across that frame would score a 0. Our current method describes narrative similarity as a proxy for character relation similarity; this extension will enrich that description. Next stages for this research include refining the comparison algorithm, applying it to a corpus of dissimilar narratives, implementing the role of the hypernym in event comparisons, and assessing the method's ability to cluster stories by narrative similarity.

Acknowledgements. This work is supported in part by NSF award 1209172.

References

1 Mieke Bal. *Narratology: Introduction to the theory of narrative.* University of Toronto Press, 1997.

2 Clara Doty Bates. *Little Red Riding-Hood.* D. Lothrop and Co., 1883.

3 Steven Bethard and James H Martin. Learning semantic links from a corpus of parallel temporal and causal relations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 177–180. Association for Computational Linguistics, 2008.

4 Vincent D. Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM review*, 46(4):647–666, 2004.

5 Branimir Boguraev and Rie Kubota Ando. Timeml-compliant text analysis for temporal reasoning. In *IJCAI*, volume 5, pages 997–1003, 2005.

6 Angela Carter. The werewolf. In *The Bloody Chamber*, pages 108–110. Penguin Books, 1993.

7 Angel X Chang and Christopher D Manning. Sutime: A library for recognizing and normalizing time expressions. In *LREC*, pages 3735–3740, 2012.

8 Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.

9 Mark Alan Finlayson. *Learning narrative structure from annotated folktales.* PhD thesis, Massachusetts Institute of Technology, 2012.

10 Bernhard Fisseni and Benedikt Löwe. What Makes Stories Similar? Report on a Research Project, 2011-2014 (Invited Report). In Mark A. Finlayson, Jan Christoph Meister, and Emile G. Bruneau, editors, *2014 Workshop on Computational Models of Narrative*, volume 41 of *OpenAccess Series in Informatics (OASIcs)*, pages 9–12, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

11 Gérard Genette. *Narrative discourse: An essay in method.* Cornell University Press, 1983.

12 Jacob Grimm and Wilhelm Grimm. Rotkäppchen. In *Kinder- und Hausmärchen*, pages 113–118. Realschulbuchhandlung, v. 1, no. 26 edition, 1812. translated by D. L. Ashliman.

13 Marti A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64, 1997.

14 Marianne W. Jørgensen and Louise J. Phillips. *Discourse analysis as theory and method*. Sage, 2002.

15 Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

16 Giorgos Kollias, Shahin Mohammadi, and Ananth Grama. Network similarity decomposition (nsd): A fast and scalable approach to network alignment. *Knowledge and Data Engineering, IEEE Transactions on*, 24(12):2232–2243, 2012.

17 Andrew Lang. Little red riding hood. In Andrew Lang, editor, *The Blue Fairy Book*, pages 51–53. Longmans, Green, and Company, 5th edition, 1891.

18 Hector Llorens, Estela Saquete, and Borja Navarro-Colorado. Timeml events recognition and classification: learning crf models with semantic roles. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 725–733. Association for Computational Linguistics, 2010.

19 David M. Magerman. Statistical decision-tree models for parsing. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 276–283. Association for Computational Linguistics, 1995.

20 Olivia March and Timothy Baldwin. Automatic event reference identification. In *Australasian Language Technology Association Workshop 2008*, volume 6, pages 79–87, 2008.

21 Charles Marelles. The true history of little golden-hood. In *The Red Fairy Book*, pages 215–219. Longmans, Green, and Company, 1895.

22 Ben Miller, Ayush Shrestha, Jason Derby, Jennifer Olive, Karthikeyan Umapathy, Fuxin Li, and Yanjun Zhao. Digging into human rights violations: Data modelling and collective memory. In *Big Data, 2013 IEEE International Conference on*, pages 37–45. IEEE, 2013.

23 George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

24 Beatrix Potter. *The Tale of Jemima Puddle-Duck*. Frederick Warne and Company, 1908.

25 Vladimir Iakovlevich Propp. *Morphology of the Folktale*, volume 9 of *Publications of the American Folklore Society. Bibliographical and special series*. University of Texas Press, 1968.

26 Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. Evita: a robust event recognizer for qa systems. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 700–707. Association for Computational Linguistics, 2005.

27 Christian Schneller. *Märchen und Sagen aus Wälschtirol: Ein Beitrag zur deutschen Sagenkunde*, chapter Das Rothhütchen, pages 9–10. Verlag der Wagner'schen Universitäts-Buchhandlung, 1867. Translated by D. L. Ashliman.

28 Ayush Shrestha, Ying Zhu, Ben Miller, and Yi Zhao. Storygraph: Telling stories from spatio-temporal data. In *Advances in Visual Computing*, pages 693–702. Springer, 2013.

29 Jamshid J. Tehrani. The phylogeny of little red riding hood. *PloS one*, 8(11):e78871, 2013.

30 James Thurber. The little girl and the wolf. In *Fables for Our Time and Famous Poems Illustrated*, page 3. Harper Collins, 1983.

31 Sara Tonelli, Rachele Sprugnoli, and Manuela Speranza. Newsreader guidelines for annotation at document level. Technical report, Technical Report NWR-2014-2, Fondazione Bruno Kessler, 2014.

32 Marc Verhagen, Inderjeet Mani, Roser Sauri, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, and James Pustejovsky. Automating temporal

annotation with tarsqi. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 81–84. Association for Computational Linguistics, 2005.

33 A. H. Wratislaw. Little red hood. In *Sixty Folk-Tales from Exclusively Slavonic Sources*, pages 97–100. Elliot Stock, no. 15 edition, 1889.

Towards Narrative-Based Knowledge Representation in Cognitive Systems

Nicolas Szilas

TECFA, FPSE
University of Geneva
CH 1211 Genève 4, Switzerland
Nicolas.Szilas@unige.ch

Abstract

The hypothesis according to which narrative is not only a prominent form of human communication but also a fundamental way to represent knowledge and to structure the mind has been limitedly but increasingly discussed for the last 40 years. However, in the realm of Artificial Intelligence, it did not lead to an elaborate model of knowledge representation, beyond scripts and cases. In this paper, we attempt to go further by identifying three differentiating features of narratives that may inspire novel forms of knowledge representation: transformation, conflict and unactualized events. In particular, these three features open the way for knowledge representation formalisms that take greater account of the co-existence of intertwined conflicting representations, with various validities and validity domains, beyond a purely factual representation of the world.

1998 ACM Subject Classification I.2 Artificial Intelligence

Keywords and phrases cognitive science, narrative theories, knowledge representation

Digital Object Identifier 10.4230/OASIcs.CMN.2015.133

1 The narrative hypothesis in cognition

Cognitive science and narrative theory have developed separately, with limited dialogue between the 1950s and the 1990s, as illustrated by the absence of the entry "narrative" in the *MIT Encyclopedia of the Cognitive Sciences* [10]. These two large domains have both emerged from the need to combine various points of views from distinct disciplines with the goal of studying cognition and narrative respectively. Whereas cognitive science has covered psychology, neuroscience, epistemology, computer science and linguistics, narratology has covered literature studies, anthropology, sociology and linguistics.

However, from the 1990s the two "interdisciplines" have initiated a dialogue, in which two symmetrical directions of influence can be observed [10, 27]: How cognitive science could provide relevant models of narrative, in terms of reader's modeling (cf. cognitive narratology); and how narrative could provide relevant models of cognition, in terms of interpreting the world and reasoning about it. The focus of this article will be put on the latter, that is, the processing of information in narrative terms.

There has been extensive research on text comprehension, focusing on how a text, often a narrative text, is processed and represented as a mental structure. Such models include hierarchical decomposition via grammars [17, 36], a configuration of plot units – small patterns of affective states – [16], causal network [37], and many others. This body of research has focused exclusively on structures that represent a narrative discourse provided as a text.

In contrast, J. Bruner has significantly broadened the scope of narrative in his influential article: "The narrative construction of reality" [6]. In this paper, Bruner argues that in

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 133–141
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

order to make sense of human interaction, our mind needs to be narratively structured: "we organize our experience and our memory of human happenings mainly in the form of narrative". For Bruner, narrative is not discussed as a prominent universal form of human communication but as a form of knowledge representation for a large class of situations in the world, not just storytelling situations per se. In this vein, D. Herman states in his search for a "Story Logic" within the human mind: "narrative constitutes a logic in its own right, providing human beings with one of their primary resources for organizing and comprehending experience" [11]. However, in the rest of the discussion, Herman tends to step back to the understanding of narrative discourse, as does his subsequent book entitled "Story Logic" [11]. R. Schank adopts a wider scope when stating that "stories about one's experiences, and the experiences of others, are the fundamental constituents of human memory, knowledge, and social communication" [29], in the sense that any experience would be coded as stories, not as facts. We concern with such a larger view stating that narrative is a logic for structuring the experience in general, not just story-like inputs. In other words, from our point of view, it is worth studying whether a non-narrative text or a non-narrative experience is still processed in a narrative way. If a cognitive system such as the human mind tends to construct a narrative from any real-life experience, then the story structures evoked above in the domain of narrative text comprehension would be candidate for a general knowledge representation approach in cognition. Finally, while Bruner appears to focus on the "messy domain of human interaction", we propose to discard such a restriction and claim that narrative is a way to understand a still larger class of phenomena. In particular, by the effect of personification, many objects and events can be attributed two fundamental properties of narrative: character and intention [26]. Importantly, a narrative-based representation is not static but possibly ongoing long after the exposure of stimuli, in an attempt to reconstruct one or more representations that fit the experience.

In the rest of the paper, we call the hypothesis that narrative should be used to interpret a large class of real-world happenings **the narrative hypothesis**. This hypothesis is speculative and has been criticized by M.-L. Ryan [27]. However, we are not convinced by her demonstration, because it postulates that narrative is the result of various abilities such as experiencing emotions, having a sense of chronological ordering, being able to infer causal relations. However, the narrative hypothesis states that these abilities do not come first but *with* narrative, as it will be detailed below. Based on the narrative hypothesis, we form two research questions:

1. Has the narrative hypothesis been used in the field of Artificial Intelligence (AI)?
2. If not, or not much, how and for what purpose should we use it?

Through these questions we tend to explore that if AI manages to draw valuable computational techniques from the narrative hypothesis then this hypothesis will acquire some validity and make narrative studies a genuine contributor to cognitive science.

2 AI for Narrative, Narrative for AI

In the field of AI, we are interested in the domain of Knowledge Representation (KR). Our question in this context is: Is there a KR technology that is based on the narrative hypothesis? R. Davis his colleagues [8] consider five different roles for any knowledge representation: 1) as a surrogate, 2) as a set of ontological commitments, 3) as a tool of reasoning, 4) as a medium for efficient computation and 5) as a medium of human expression. Therefore, our question is: Is there a KR that has, as a fundamental way to view the world, the narrative hypothesis (ontological commitment)?

A large variety of KR approaches have been proposed in cognitive science: rules, frames, scripts [28], semantic nets, cases, conceptual graphs [31], etc.. Two of them have been found to share similarities with the narrative hypothesis: scripts and cases. As KR, scripts and cases contrast with logic-based approaches in the sense that they no longer consider reasoning solely as logic deduction process, but also as storage of stereotypical situations that embed a known solution. For *scripts*, this situation includes "a predetermined stereotyped sequence of actions" [28], which resembles a story. Schank and Abelson propose that our memory is constituted of many of these scripts. They guide our understanding of both narrative text and real-world events, by being first recognized as appropriate and then used (after possible adaptation) in the current situation. For *cases*, what is stored is not necessary a story-like structure as for scripts, but a problem-solution couple that corresponds to a case that has been successfully solved previously. Contrary to scripts, cases have been widely used in the field of AI to solve a large range of problems. However, scripts and cases cover minimally the notion of narrative. As Schank and Abelson state, "a script is, in effect, a very boring little story" [28]. Scripts share with narrative the idea of temporal succession and character, but the former lack many other features such as intention (stored outside the script), emotion, conflict, evaluation, and closure. In that sense, they do not constitute the narrative construction of reality called by Bruner [6]. Besides, there has been a significant increase in computational models of narrative research in the field of Interactive Storytelling since the late 1990's. With the goal of generating narratives (in various media including 3D worlds) or driving narrratively the experience in an interactive narrative such as an adventure video game, this field has produced a wide range of narrative models based on various narrative principles: Aristotelian/Freytagian tension curve [18], characters' intentions [2, 7], characters' emotions [2], audience's emotional response [32, 41], dilemma [3, 34], conflict [33, 40], causality [22, 24], etc. Although these models of narrative were not conceived as models of cognition, we raise the question whether some of them, once adapted, could play such a role.

In the rest of the paper, we will explore this possibility by first defining more precisely the requirements for a narrative-based KR and then by proposing some routes for such a model.

3 From knowledge to stories . . . or reverse!

Before studying the requirements for a narrative-based KR, it is necessary to precise our viewpoint regarding the positioning of narrative in terms of level of processing. From a cognitive perspective, the ability to process narratives has often been considered as a high level feature of cognition. For example, in early structuralist narratology, narrative goes "beyond the sentence" and constitutes a "large sentence" [4], which implicitly means that one needs to be able to make and understand sentences (covered by the field of linguistics) before being able to make and understand narratives. In a totally different narratological tradition, Labov and Waletzky [14], studying oral narratives, define narrative as "one method for recapitulating past experience by matching a verbal sequence of clauses to the sequence of events which actually occurred". This definition presupposes that the events must initially happen and be stored before being later processed narratively, which is in contrast with the above-mentioned narrative hypothesis stating that narrative is the way the events are encoded. Finally, the question raised by the present conference "Can narrative be subsumed by current models of higher-level cognition, or does it require new approaches?" has positioned narrative as a higher-level cognitive phenomenon. We challenge this position in suggesting that, as a hypothesis, narrative should be a basic and primitive way to process and store information.

While one tends to see narrative as made of characters, goals, values, etc., we suggest that the latter elements may be build as an outcome of a more fundamental and narrative-based representation. As Schank and Abelson put it in a somewhat extreme statement: “We propose that there is no factual knowledge as such in memory” [29]. This primacy of narrative is consistent with B. Victorri’s views on the relation between linguistics and narrative [38]. He claims that language would be the result of narrative, making it possible for human beings to survive by recalling a past experience, which is contrary to the linguistics’ point of view – narrative is considered to be a by-product of language and language is used to give true/false statements about the world. It is naturally out of the scope of this research to discuss such a hypothesis, but it illustrates that the “natural” ordering of things – first we represent objects and their relation and second we make a story out of it – may be an illusion.

From a computational point of view, AI comes from logic and symbolic reasoning. This has been intensively challenged by connectionism who raised the question on how these symbols appeared in the human mind with an emphasis on learning by the adjustment of continuously-valued units [30]. In our case, the logico-symbolic is criticized in a less radical way: we suppose that there exists an intermediate narrative representation between a simple episodic memory and higher-level symbols. In other words, instead of storing “the glass is on the table” that can be represented by various KR approaches, we would store a narrative representation stemming from the experience of putting a glass on a table and observing with surprise that it did not fall. Compared to Schank and Abelson position however, we are not claiming that “The mind can be seen as a collection of stories, collections of experiences one has already had” [29] because this intermediate narrative KR may be (and certainly is) an abstraction of these stories. This narrative representation may be closer to recent work on counterfactual reasoning [20]. In addition, it would be interconnected with other forms of representation, forming a hybrid representation/system, a known research domain in AI.

Back to interactive storytelling research, the absence of such an intermediate KR may explain why “Early on, artificial intelligence researchers showed that enormously complex linguistic and cognitive operations are required to generate or comprehend even the most minimal stories.” [11, p. 1]. AI researchers may simply have used the wrong tools to generate stories in attempting to reconstitute them from symbolic factual descriptions of the world’s entities, while they may have been advantageously described via on a more suited KR.

4 Narrative Features for KR

4.1 Approach

While we have identified the lack of a KR corresponding to the narrative hypothesis, the question of the utility of such a KR must be raised. In terms of the above-mentioned five roles identified by Davis and colleagues [8], two roles are missing: as a tool of reasoning and as a medium for efficient computation. That is, one needs to identify, from a computational point of view, which advantages would bring a narrative representation of the world. In the following parts, instead of proposing a fully specified KR approach, we investigate which narrative-specific feature of narrative could be used for building a narrative-based KR. J. Bruner argues that memory is structured narratively and enumerates ten features of narrative that he judges as particularly relevant to examine “how [narrative] operates as an instrument of mind in the construction of reality” [6]. D. Hermann, in his quest for “narrative as an instrument of mind” identifies “five ways stories scaffold intelligent behaviors” [12]: chunking experience, imputing causal relations, problem raising/solving, sequencing actions, distributing intelligence. Our approach is slightly different because we want to push the narrative hypothesis further by targeting a specific and useful form of KR. Therefore we need to identify more precise narrative features. For instance, chunking experience and imputing causal relation are not specific to narrative. Similarly, sequencing of actions is not sufficient to characterize narrative, if we admit, with J.-M. Adam that a cooking recipe is not a story [1]. We are focusing in the following on three essential narrative features in hoping that they are the differentiating bedrocks for a future narrative-based KR.

4.2 Narrative transformation and Knowledge Acquisition

One of the fundamental characteristics of narrative is the transformation that underlies any story. Transformation is part of several definitions of narrative [1, 26]. This transformation concerns the heroes of the story and more importantly it concerns the audience as well. From the pragmatics' viewpoint, narrative is a form of discourse that carries a message from the author to the audience [1]. Experiencing a narrative is a form of knowledge acquisition, which is based on various strategies that include storage of story events in the episodic memory, transmission of factual information regarding the world (the fictional world is never totally disconnected from the real world), transmission of a moral viewpoint through the story's value system [13]. Therefore, a cognitive system using a narrative-based KR does not store knowledge solely as a static representation but as the transformation that leads to that knowledge. This is a fundamental change compared to traditional KR that aims at representing the world in a static and unambiguous manner. Conversely, relating a given knowledge to a past and possibly erroneous knowledge is in line with the constructivist epistemology. The constructivist epistemology states that if older knowledge may be false compared to newer knowledge, it is still valid and useful in restricted domains of validity – the classical example in the history of science being the Newtonian mechanics, invalidated by the theory of relativity, but still useful in everyday calculation. A narrative-based KR would be able to relate different pieces of knowledge, by linking newly acquired knowledge and previous knowledge that it is supposed to supersede. From an AI perspective, such a KR would allow not only to keep and use knowledge that is generally wrong but applicable within its domain of validity, but also to identify the domains of validity and invalidity via the stories attached to the successively acquired knowledge. This is related to the notion of context.

4.3 Dramatic conflict and cognitive conflict

Around the term "conflict", there is a striking similarity, at least in terminology, between narrative (drama in particular) and learning. In dramaturgy, conflict is recognized as a key mechanism of drama[1], a principle largely used within the screenwriting community, via the motto "All drama is conflict" [9, p. 24]. It is a term with a broad meaning, that may include antagonism between characters, physical (or external) obstacles, and internal dilemma [15, 19]. In constructivist learning theory, cognitive conflict plays a key role in bringing a learning subject to change his/her internal representation in order to accommodate new information from the world [21]. Cognitive conflict is an incompatibility between the subject's representations and new facts. The subject may reject the new fact because of the conflict or search for a new representation that would integrate the fact. Based on an analogy between these two conflicts, how could a narrative view on KR provide a suited

[1] This principle is sometimes wrongly attributed to Aristotle, but it rather seems to emerge in the XIXth century.

model for knowledge acquisition? There is no straightforward answer since the notion of conflict in narrative can be interpreted in various ways when it comes to implement it in a computational model [32, 39]. We will offer an initial level of answer with consideration of the following stereotypical proto-story: In a certain situation, character C wants to reach a goal G by attempting an action A that, according to his current knowledge, must lead to G. However, without any external intervention, action A leads to another situation and G is not reached. C is puzzled and looks for an explanation that he find later in the story. This story embeds an obstacle, a typical dramatic element that is a sort of dramatic conflict, maybe not the most interesting, and generates an emotional response: the surprise of the character as well as his disappointment, both leading to an emotional response of the audience, via the mechanism of empathy [35]. While this story falls below the sophistication of many simple stories, it is still more narrative than scripts as described above, since it embeds conflict and emotion. Furthermore, this story tells how certain knowledge has proven wrong and how it could be replaced by a new knowledge. A narrative-based KR could store the fundamental conflict of the above story within the acquired knowledge. Then, not only, as we discussed above, would the knowledge be supplemented with the previous knowledge it supersedes, but also would it embed the elements that characterize a conflicting situation between knowledge and the emotional valence attached to that situation. What is embedded is not the story itself (the sequence), but an abstraction that codes the core conflictual elements in the story. Such abstractions have been proposed in interactive storytelling research [3, 32, 5].

4.4 The disnarrated, the unactualized and the hypothetical reasoning

Because narrative is often defined as telling events that have certain characteristics, a dimension of narrative is often neglected: events that do not occur in the fabula or events that are not narrated, G. Prince called the latter the disnarrated [23]. It covers many types of events: ellipses, events that by their nature are difficult to tell [23], hypothetical events in possible worlds [25], counterfactual events, etc. In the above-mentioned epistemological point of view, some unactualized events correspond to what could have occurred if a given knowledge were true, while it did not occur because this knowledge was not true in this context. This is illustrated for example in the following excerpt: "The slightest breeze that ruffles the surface of the water makes you bow your heads, while I, the mighty Oak, stand upright and firm before the howling tempest."[2]. The following of the story proves this affirmation wrong. The disnarrated events and the unactualized events correspond in fact to an essential feature of the hypothetico-deductive scientific methodology: elaborating of an experimental setting where two results could occur with one validating the hypothesis and thus promoting a new knowledge and the other invalidating the hypothesis and leading to a status-quo. In the above proto-story, the unreached goal G is disnarrated or narrated in a conditional mode – the consequences of its reaching do not occur – but it is still part of the story. Therefore, this suggests that a narrative-based KR would naturally and natively include the disnarrated and unactualized events. For example, the knowledge formulated as a fact by "The earth is round" can be narratively represented by "A person travels straightforward to reach the end of the earth, but he does not reach this end. He finally reaches his starting point". Another example, the fact "birds fly with their wing" may be narratively represented by a story with a farmer clipping the wings of his chicken (although this example is misleading, since chicken cannot really fly). This is not a common way to

[2] From the Aesop's fable "The Oak and the Reeds".

represent knowledge in AI, but in addition to be more psychologically plausible, it may prove useful in knowledge-based systems to provide explanation of the outputs.

5 Conclusion, future work

Following the studies of J. Bruner, R. Schank and D. Hermann, we have explored how narrative could be viewed as a fundamental way to represent knowledge. Our goal is to go further in designing and implementing a computational model of narrative, not for processing narratives (generation or analysis) but to represent knowledge in a much broader scope. While this ambitious goal has not been reached yet, our intention with this contribution was first to identify it and present it to the research community, as a new direction in AI within the broad umbrella of Cognitive Science. In the spirit of the latter, two main directions of research could be followed. The first direction consists in validating a narrative-based KR model via psychological experimentation. This involves inventing an experimental protocol showing that non-narrative information is stored in a narrative manner, rather than as declarative knowledge. By "in a narrative manner", one needs to understand more than "sequentially" or "procedurally": typical narrative elements such as conflict, suspense, evaluation need to be there. The second direction consists in designing and implementing a computational model of KR that is different and, for some purposes, more powerful than existing KR approaches. We have not yet identified what task such a KR model should help to accomplish, which constitutes a future challenge of this research. In terms of computational model, it may be an extension of Case-Based Reasoning, where "correct" cases and "incorrect" cases would co-exist in a conflictual manner; Or it may be an advanced explanation system for a knowledge base; Or it may be a hybrid system, combining a rule-based system with a narrative-based system, each with its own inference mechanism. The complexity and richness of narrative may open many fresh directions in AI, revigorating the dialog between computational intelligence and human intelligence, in the tradition of Cognitive Science.

References

1 Jean-Michel Adam. *Le texte Narratif.* Nathan, Paris, 1994.

2 Ruth Aylett, Sandy Louchart, Joao Dias, Ana Paiva, Marco Vala, Sarah Woods, and Lynne Hall. Unscripted narrative for affectively driven characters. *IEEE Journal of Graphics and Animation*, 26(May/June):42 – 52, 2006.

3 Heather Barber and Daniel Kudenko. Dynamic generation of dilemma-based interactive narratives. In *Proc. Third Conf. on Artificial Intelligence and Interactive Digital Entertainment – AIIDE*, pages 2–7, Menlo Park, CA, 2007. AAAI Press.

4 Roland Barthes. Introduction à l'analyse structurale des récits. *Communications*, 8(1):1–27, 1966.

5 Cristina Battaglino, Rossana Damiano, and Vincenzo Lombardo. Moral Values in Narrative Characters: An Experiment in the Generation of Moral Emotions. In David Oyarzun, Federico Peinado, R. Michael Young, Ane Elizalde, and Gonzalo Méndez, editors, *ICIDS*, pages 212–215, Heidelberg, 2014. Springer.

6 Jerome Bruner. The narrative construction of reality. *Critical Inquiry*, 18:1–21, 1991.

7 Marc Cavazza, Fred Charles, and Steven J. Mead. Characters in Search of an author: AI-based Virtual Storytelling. In Olivier Balet, Gérard Subsol, and Patrice Torguet, editors, *International Conference on Virtual Storytelling (ICVS 2001). LNCS 2197*, Lecture Notes in Computer Science, pages 145–154. Springer, Heidelberg, September 2001.

8 Randall Davis, Howard Shrobe, and Peter Szolovits. What is a Knowledge Representation? *AI Magazine*, 14(1):17–33, 1993.

9 Syd Field. *Screenplay – The Foundations of Screenwriting.* Dell Publishing, New York, 1984.

10 David Herman. Narratology as a cognitive science. *Image [&] Narrative, online Magasine of the Visual Narrative*, 1(1), 2000.

11 David Herman. *Story Logic: Problems and Possibilities of Narrative.* University of Nebraska Press, Lincoln, 2002.

12 David Herman. *Storytelling and the Sciences of Mind.* MIT press, 2013.

13 Vincent Jouve. *Poétique des valeurs.* PUF, coll. "Ecriture", Paris, 2001.

14 William Labov and Joshua Waletzky. Narrative analysis: Oral versions of personal experience. *Essays on Verbal and Visual Arts*, pages 12–44, 1967.

15 Yves Lavandier. *La dramaturgie.* Le clown et l'enfant, Cergy, France, 1997.

16 Wendy Lehnert. Plot units and narrative summarization. *Cognitive Science*, 5(4):293–331, December 1981.

17 Jean M. Mandler and Nancy S. Johnson. Remembrance of things parsed: Story structure and recall. *Cognitive Psychology*, 9(1):111–151, 1977.

18 Michael Mateas and Andrew Stern. Integrating Plot, Character and Natural Language Processing in the Interactive Drama Façade. In Stefan Göbel, Norbert Braun, Ulrike Spierling, Johanna Dechau, and Holger Diener, editors, *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, AAAI Fall Symposium Series, pages 139–151, Darmstadt, 2003. Fraunhofer IRB.

19 Robert McKee. *Story: Substance, Structure, Style, and the Principles of Screenwriting.* Harper Collins, New York, 1997.

20 Srini Narayanan. Mind Changes: A simulation semantic model of counterfactuals. mind changes, 2012. URL: `http://www1.icsi.berkeley.edu/~snarayan/counterfactuals.pdf`.

21 Jean Piaget. *L'équilibration des structures cognitives: probème central du développement*, volume 33. Presses universitaires de France, 1975.

22 Julie Porteous, Marc Cavazza, and Fred Charles. Applying planning to interactive storytelling: Narrative control using state constraints. *ACM Transactions on Intelligent Systems and Technology*, 1(2):10:1–10:21, 2010.

23 Gerald Prince. The Disnarrated. *Style*, 22(1):1–8, 1988.

24 Mark Riedl, C J Saretto, and R Michael Young. Managing Interaction Between Users and Agents in a Multi-agent Storytelling Environment. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 741–748, New York, NY, USA, 2003. ACM.

25 Marie-laure Ryan. *Possible Worlds, Artificial Intelligence, and Narrative Theory.* Indiana University Press, Bloomington, IN, 1991.

26 Marie-Laure Ryan. Introduction. In Marie-Laure Ryan, editor, *Narrative Across Media.* University of Nebraska Press, Lincoln and London, 2004.

27 Marie-laure Ryan. Narratology and Cognitive Science: a Problematic Relation. *Style*, 44(4):469–495, 2010.

28 Roger C Schank and Robert P Abelson. Scripts, Plans, and Knowledge. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'75, pages 151–157, San Francisco, CA, USA, 1975. Morgan Kaufmann Publishers Inc.

29 Roger C Schank and Robert P Abelson. Knowledge and memory: The real story. *Knowledge and memory: The real story. Advances in social cognition*, 8:1–85, 1995.

30 Paul Smolensky. Connectionist AI, symbolic AI, and the brain. *Artificial Intelligence Review*, 1(2):95–109, 1987.

31 John F Sowa. Conceptual Graphs for a Data Base Interface. *IBM Journal of Research and Development*, 20(4), 1976.

32 Nicolas Szilas. A Computational Model of an Intelligent Narrator for Interactive Narratives. *Applied Artificial Intelligence*, 21(8):753–801, 2007.

33 Nicolas Szilas, Jason Barles, and Manolya Kavakli. An implementation of real-time 3D interactive drama. *Computers in Entertainment*, 5(1):5, January 2007.

34 Nicolas Szilas and Urs Richle. Towards a Computational Model of Dramatic Tension. In Mark A Finlayson, Bernhard Fisseni, Benedikt Löwe, and Jan Christoph Meister, editors, *2013 Workshop on Computational Models of Narrative*, volume 32 of *OpenAccess Series in Informatics (OASIcs)*, pages 257–276, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://dx.doi.org/10.4230/OASIcs.CMN.2013.257`.

35 Ed Tan. *Emotion and the structure of narrative film. Film as an emotion machine.* Erlbaum, Mahwah, NJ, 1996.

36 Perry W Thorndyke. Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, 9(1):77–110, 1977.

37 Tom Trabasso, Paul Van Den Broek, and So Young Suh. Logical necessity and transitivity of causal relations in stories. *Discourse Processes*, 12(1):1–25, 1989.

38 Bernard Victorri. La place de la fonction narrative dans l'émergence du langage et la structure des langues. *Théorie, Littérature, Enseignement*, 1(17):23–38, 1999. URL: `https://halshs.archives-ouvertes.fr/halshs-00009329`.

39 Stephen G Ware and R Michael Young. CPOCL: A Narrative Planner Supporting Conflict. In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, number Young in AAAI Proceedings, pages 97–102, Palo Alto, CA, 2011. AAAI Press.

40 Stephen G Ware, R Michael Young, Brent E Harrison, and David L Roberts. Four Quantitative Metrics Describing Narrative Conflict. In David Oyarzun, Federico Peinado, R Michael Young, Ane Elizalde, and Gonzalo Méndez, editors, *Fifth International Conference on Interactive Digital Storytelling (ICIDS). LNCS, 7648*, LNCS, pages 18–29, Heidelberg, 2012. Springer.

41 Peter Weyhrauch. *Guiding Interactive Drama.* PhD thesis, Carnegie Mellon University, 1997.

Governing Narrative Events With Institutional Norms

Matt Thompson[1], Julian Padget[1], and Steve Battle[2]

1 University of Bath
Bath, United Kingdom
{mrt32,masjap}@bath.ac.uk

2 Sysemia Ltd
Bristol, United Kingdom
steve.battle@sysemia.co.uk

Abstract

A narrative world can be viewed as a form of society in which characters follow a set of social norms whose collective function is to guide the characters through (the creation of) a story arc and reach some conclusion. By modelling the rules of a narrative using norms, we can govern the actions of agents that act out the characters in a story. Agents are given sets of permitted actions and obligations to fulfil based on their and the story's current situation. However, the decision to conform to these expectations is ultimately left to the agent. This means that the characters have control over fine-grained elements of the story, resulting in a more flexible and dynamic narrative experience. This would allow the creator of an interactive narrative to specify only the general structure of a story, leaving the details to the agents. We illustrate a particular realisation of this vision using a formalization of Propp's morphology in a normative social framework, with belief-desire-intention agents playing the characters.

1998 ACM Subject Classification I.2.11 Distributed Artificial Intelligence

Keywords and phrases institutions, norms, narrative, agents

Digital Object Identifier 10.4230/OASIcs.CMN.2015.142

1 Introduction

A satisfying narrative must be more than just a series of interactions between character agents. There is a need for some underlying structure to these interactions. Additionally, agents are not a natural way to model events such as off-screen occurrences or scene introductions from a narrator.

Simulating a narrative using intelligent agents as characters offers many advantages. Each agent can be programmed to behave in certain idiosyncratic ways, based on a psychological or behavioural model. A common approach to add narrative structure to an agent-based simulation is to implement a drama manager, as in Mateas and Sterns' *Façade* [9].

This presents a problem: if the agents are being governed by a drama manager, to what extent are they autonomous? Do they still have some degree of 'free will' to carry out their own individual actions, in accordance with their personalities?

Other approaches to balancing authorial control with player or character agency include the use of director agents [8], reincorporation of player actions back into the narrative [15] and mediation to prevent narrative-breaking actions [12].

In this paper we present an approach to regulating narrative structure while still allowing agents some degree of autonomy. The narrative world is described and managed using an institutional model.

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 142–151
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

An institutional model can be thought of as a model of society. By specifying a set of social norms, certain agent behaviours can be encouraged or discouraged according to the needs of that society. Institutions have been used to simulate the workings of auctions [3], vehicle convoys [1] and crowd movement [7]. All these applications are similar in that they all involve intelligent agents working together in a social environment.

The advantages of using institutions to govern agents' behaviours is that they still allow the agents some autonomy in their actions. The rules of a society are implied, and while adherence to these rules is encouraged, it is possible for them to be broken (often incurring a penalty). This makes them ideal for regimenting the actions of characters in a narrative. In order to have a narrative that is satisfying and consistent with a certain story world, some kind of structure is needed. However, if this narrative is to be interactive, the characters within the narrative need some degree of freedom in their actions. They need the ability to bend or break the rules of the storyworld at times, in order to surprise the player. Institutions make this possible for the agents to do. However, as with breaking the rules of any society, diverging from the norm may bring penalties and hardship upon the deviating agent.

In order to describe a narrative using an institution, we use Vladimir Propp's formalism of Russian folktales, from "The Morphology of the Folktale" [10].

2 Propp's Morphology of the Folktale

Propp's seminal work "The Morphology of the Folktale" [10], though first published in 1928, is still a widely-used formalism for researchers and game designers looking to generate narratives procedurally. Propp identifies recurring characters and motifs in Russian folklore, distilling them down to a concise syntax with which to describe stories.

In this formalism, characters have *roles*, such as *hero*, *villain*, *dispatcher*, *false hero*, and more. Characters performing a certain role are able to perform a subset of *story functions*, which are actions that make the narrative progress. For example, the *dispatcher* might send the *hero* on a quest, or the *victim* may issue an *interdiction* to the *villain*, which is then *violated*.

Propp defines a total of 31 distinct story functions, some of which can have subtle variations from story to story. Each function is given a number and symbol in order to create a succinct way of describing entire stories. Examples of such functions are:

- One of the members of a family absents himself from home: *absentation*.
- An interdiction is addressed to the hero: *interdiction*.
- The victim submits to deception and thereby unwittingly helps his enemy: *complicity*.
- The villain causes harm or injury to a member of the family: *villainy*.

Each of these functions can vary to a great degree. For example, the *villainy* function can be realised as one of 19 distinct forms of villainous deed, including *the villain abducts a person*, *the villain seizes the daylight*, and *the villain makes a threat of cannibalism*.

These functions are enacted by characters following certain roles. Each role (or *dramatis personae* in Propp's definition) has a *sphere of action* consisting of the functions that they are able to perform at any point in the story. Propp defines seven roles that have distict spheres of action: *villain*, *donor*, *helper*, *princess*, *dispatcher*, *hero*, and *false hero*.

In a typical story, one story function will follow another as the tale progresses in a sequential series of cause and effect. However, Propp's formalism also allows for simultaneous story functions to occur at once.

2.1 Example: A Punch and Judy show

Consider the classic British-Italian "Punch and Judy" puppet show often seen at English seaside resorts. The "Punch and Judy" world is a very simple and consistent narrative domain, in which simplistic characters act out predictable sequences of events. The key features of a Punch and Judy show include:

- The show is introduced by a clown named "Joey".
- Punch beats and kills his child, and then his wife Judy.
- There is a scene where Punch chases a monkey or cat.
- A policeman tries to arrest Punch, but is instead killed by him.
- Joey asks Punch to look after some sausages in one scene. Shortly after Joey leaves, a crocodile appears and eats them.
- Punch, the lead character, beats and kills almost every other character by the end of each scene. Only Joey and sometimes the monkey or cat avoid this fate.
- The show sometimes ends with an encounter between Punch and the Devil, which Punch wins.

Despite this harrowing combination of narrative elements, Punch and Judy is considered a *farce* due to the over-the-top violence and simplicity of its world. It is usually performed as a puppet show for children, who are encouraged to cheer or boo the puppets.

The common elements of Punch and Judy are easily described in terms of Propp's story functions. Using the example where Joey asks Punch to guard some sausages, the appropriate story functions are:

1. Joey tells Punch to look after the sausages (*interdiction*).
2. Joey has some reservations, but decides to trust Punch (*complicity*).
3. Joey gives the sausages to Punch (*provision or receipt of a magical agent*).
4. Joey leaves the stage (*absentation*).
5. A crocodile enters the stage and eats the sausages (*violation*).
6. Punch fights with the crocodile (*struggle*).
7. Joey returns to find that the sausages are gone (*return*).

In order to better model the Punch and Judy world in terms of Propp functions, we have allowed some flexibility of the roles that each agent assumes. At points Punch is the hero, at other times he is the villain. Sometimes Joey is the hero, but he can also be a donor (a character who gives an object to the hero). The crocodile is a villain, but other characters are all certainly victims (since they are all obliged to be killed by Punch as part of the Punch and Judy story world).

One novel aspect of managing these Propp functions with an institutional model is that the agents' roles *can* be flexible. If the audience cheers on Judy as she hits Punch, why not fulfil their desires and make her the hero, and Punch the victim? This is what we aim to achieve with our approach: a story world where certain rules do hold, but are flexible enough to be broken if the player or audience wills it.

3 Institutions for narrative regulation

3.1 Institutions and norms

Early examples of institutional models suggest their application to the regulation of systems involving multiple actors. Noriega's "fish market" thesis describes the application of an agent-mediated institution for regulating a fish market auction scenario [3], checking the validity of agent actions and addressing the issue of agent accountability in an auction environment. Rodriguez [13], and later Vázquez-Salceda [16], refine and extend Noriega's implementation of agent-mediated institutions.

However, it is Cliffe's approach of using Answer Set Programming (ASP) to specify institutions that we use here [4]. We define an institution in terms of *deontic logic*, specifying the permissions and obligations that act upon agents at any particular point in the story.

This approach alone is not enough, however. In order to effectively model a narrative using an institution and ASP, we must use a formalism for narrative that specifies which events and actions occur at certain points in the narrative. We achieve this by translating Propp's formalism of Russian folktales [10] into actions that agents are permitted or obliged to perform.

3.2 Describing institutions with deontic logic

We describe our institution using *deontic logic*, defining our model in terms of *fluents*, *events*, *powers*, *permissions* and *obligations*.

3.2.1 Fluents

Fluents are properties that may or may not hold true at some instant in time. *Institutional events* are able to *initiate* or *terminate* fluents at points in time. A fluent could describe whether a character is currently on stage, the current scene of a story, or whether or not the character is happy at that moment in time.

Domain fluents ($\mathcal{D}$) describe domain-specific properties that can hold at a certain point in time. In the Punch and Judy domain, these can be whether or not an agent is on stage, or their role in the narrative (equation 1).

Institutional fluents consist of *institutional powers*, *permissions* and *obligations*.

$$\mathcal{D} = \{\texttt{onstage, hero, villain, victim, donor, item}\} \tag{1}$$

An **institutional power** ($\mathcal{W}$) describes whether or an agent, and by extension the action they have taken, has the authority to meaningfully generate an institutional event. Using Propp as an example, a *violated interdiction* can only occur after an *interdiction* has taken place. Therefore, the institution would not be empowered to generate a *violated interdiction* institutional event if the prior *interdiction* has not yet taken place.

Institutional powers describe what events the institution is capable of bringing about. As institutional events represent Propp's story functions in our model, the institution should only be capable of generating events if they fit in the right place in the narrative. For example, a *violation* can take place only after an *interdiction* event has occurred. Punch can only violate Joey's request to guard the sausages after the request itself has happened. Equation 2 shows a list of possible empowerments, essentially a list of institutional events.

$$\begin{aligned}\mathcal{W} = \{&\texttt{pow(introduction), pow(interdiction), pow(give), pow(absentation),}\\ &\texttt{pow(violation), pow(return)}\}\end{aligned} \tag{2}$$

Permissions ($\mathcal{P}$) are external actions that agents are permitted to do at a certain instant in time. These can be thought of as the set of *socially permitted* actions available to an agent. While it is possible for an agent to perform other actions, societal norms usually prevent them from doing so.

For example, it would not make sense in the world of Punch and Judy if Punch were to give the sausages to the Policeman. It is always Joey who gives the sausages to Punch. Also, it would be strange if Joey were to do this in the middle of a scene where Punch and Judy are arguing. We make sure agents' actions are governed so as to allow them only a certain subset of permitted actions at any one time. Equation 3 shows a list of permission fluents.

$$\begin{aligned}\mathcal{P} = \{&\texttt{perm(leavestage), perm(enterstage), perm(die), perm(kill),}\\ &\texttt{perm(hit), perm(give), perm(fight)}\}\end{aligned} \tag{3}$$

Obligations ($\mathcal{O}$) are actions that agents *should* do before a certain deadline. If the action is not performed in time, a *violation event* is triggered, which may result in a penalty being incurred. While an agent may be obliged to perform an action, it is entirely their choice whether or not they actually do so. They must weigh up whether or not pursuing other courses of action is worth suffering the penalty that an unfulfilled obligation brings.

Anybody who has seen a Punch and Judy show knows that at some point Joey tells Punch to guard some sausages, before disappearing offstage. Joey's departure is modelled in the institution as the *absentation* event. It could be said that Joey has an obligation to leave the stage as part of the *absentation* event, otherwise the story function is violated. Equation 4 shows how this would be described in the institution.

$$\mathcal{O} = \{\mathrm{obl}(\texttt{leavestage, absentation, viol(absentation)})\} \tag{4}$$

3.2.2 Events

Cliffe's model specifies three types of **event**: *external events* (or 'observed events', $\mathcal{E}_{obs}$), *institutional events* ($\mathcal{E}_{instevent}$) and *violation events* ($\mathcal{E}_{viol}$). *External events* are observed to have happened in the agents' environment, which can *generate institutional events* which act only within the institional model, *initiating* or *terminating* fluents, permissions, obligations or institutional powers. An external event could be an agent leaving the stage, an agent hitting another, or an agent dying. Internal events include narrative events such as scene changes, or the triggering of Propp story functions such as *absentation* or *interdiction* (described in Section 2).

Violation events occur when an agent has failed to fulfil an obligation before the specified deadline. These can be implemented in the form of a penalty, by decreasing an agent's health, for example.

$$\begin{aligned}\mathcal{E}_{obs} = \{&\texttt{startshow, leavestage, enterstage, die, give,}\\ &\texttt{harmed, hit, fight, kill, escape}\} \qquad (5)\\ \mathcal{E}_{instact} = \{&\texttt{introduction, interdiction, give, absentation,}\\ &\texttt{violation, return, struggle, defeat, complicity,}\\ &\texttt{victory, escape}\} \qquad (6)\\ \mathcal{E}_{viol} = \{&\texttt{viol(introduction), viol(interdiction), viol(give),}\\ &\texttt{viol(absentation), viol(violation), viol(return),}\\ &\texttt{viol(struggle), viol(defeat), viol(complicity)}\\ &\texttt{viol(victory), viol(escape)}\} \qquad (7)\end{aligned}$$

3.2.3 Event Generation and Consequences

An **event generation** function, $\mathcal{G}$, describes how events (usually external) can generate other (usually institutional) events. For example, if an agent leaves the stage while the

$$\mathcal{G}(\mathcal{X},\mathcal{E}) : \langle \emptyset, \mathit{tellprotect}(\texttt{donor, villain, item})\rangle \rightarrow \{\mathit{interdiction}\} \tag{8}$$

$$\langle \{\mathit{interdiction}\}, \mathit{agree}(\texttt{villain}))\rangle \rightarrow \{\mathit{complicity}\} \tag{9}$$

$$\langle \emptyset, \mathit{give}(\texttt{donor, villain, item}))\rangle \rightarrow \{\mathit{receipt}\} \tag{10}$$

$$\langle \{\mathit{interdiction}\}, \mathit{leavestage}(\texttt{donor})\rangle \rightarrow \{\mathit{absentation}\} \tag{11}$$

$$\langle \{\mathit{interdiction}\}, \mathit{harmed}(\texttt{item})\rangle \rightarrow \{\mathit{violation}\} \tag{12}$$

$$\langle \{\mathit{interdiction}, \mathit{absentation}\}, \mathit{enterstage}(\texttt{donor}),\ \mathrm{onstage}(\texttt{villain})\rangle \rightarrow \{\mathit{return}\} \tag{13}$$

$$\langle \emptyset, \mathit{hit}(\texttt{donor, villain})\rangle \rightarrow \{\mathit{struggle}\} \tag{14}$$

$$\mathcal{C}^{\uparrow}(\mathcal{X},\mathcal{E}) : \langle \emptyset, \texttt{receipt}\rangle \rightarrow \{\mathrm{perm}(\texttt{leavestage(donor)})\} \tag{15}$$

$$\langle \{\mathit{active}(\mathit{interdiction})\}, \texttt{violation}\rangle \rightarrow \{\mathrm{perm}(\texttt{enterstage(dispatcher)})\} \tag{16}$$

$$\langle \{\mathit{active}(\mathit{absentation}), \mathit{active}(\mathit{violation})\}, \texttt{return}\rangle \rightarrow \{\mathrm{perm}(\texttt{hit(donor, villain)})\} \tag{17}$$

$$\mathcal{C}^{\downarrow}(\mathcal{X},\mathcal{E}) : \langle \emptyset, \texttt{interdiction}\rangle \rightarrow \{\mathrm{perm}(\texttt{give(donor, villain, item)})\} \tag{18}$$

$$\langle \{\mathit{active}(\mathit{interdiction})\}, \texttt{absentation}\rangle \rightarrow \{\mathrm{perm}(\texttt{leavestage(donor)})\} \tag{19}$$

$$\langle \{\mathit{active}(\mathit{interdiction})\}, \texttt{violation}\rangle \rightarrow \{\mathit{active}(\mathit{interdiction})\} \tag{20}$$

$$\langle \{\mathit{active}(\mathit{absentation}), \mathit{active}(\mathit{violation})\}, \texttt{return}\rangle \rightarrow \{\mathit{active}(\mathit{absentation})\} \tag{21}$$

Figure 1 Generation and consequence rules for Punch and Judy.

interdiction event holds, they trigger the *leavestage* event. This combination generates the *absentation* institutional event (equation 11).

Event generation functions follow a ⟨`preconditions`⟩ → {`postconditions`} format, where the preconditions are a set of fluents that hold at that time and an event that has occurred, and the postconditions are the events that are generated. They are generally used to generate internal, institutional events from external events.

Consider the Punch and Judy scenario described in Section 2.1. There are seven institutional events (story functions) that occur during this scene: *interdiction*, *complicity*, *receipt* (from Propp's *receipt of a magical agent*) *absentation*, *violation*, *struggle*, *return*. These institutional events are all generated by external events. The *interdiction* is generated when Joey tells Punch to protect the sausages. Punch agreeing amounts to *complicity*. Joey *gives* punch the sausages (*receipt*), then leaves the stage (*absentation*). The crocodile eating the sausages is a *violation* of Punch's oath, the agents fight (*struggle*), then Joey enters the stage again (*return*).

It is desirable that these story function occur in this sequence in order for a satisfying narrative to emerge. Agents may decide to perform actions that diverge from this set of events, but the institution is guiding them towards the most fitting outcome for a *Punch and Judy* world. For this reason, a currently active story function can be the precondition for event generation. For example, the *receipt* event may only be triggered if an agent externally performs a *give* action **and** if the *complicity* event currently holds (equation 10). Examples of event generation function for this scenario, complete with preconditions, are listed in equations 8 to 14 in Figure 1.

Consequences consist of fluents, permissions and obligations that are *initiated* ($\mathcal{C}^{\uparrow}$) or *terminated* ($\mathcal{C}^{\downarrow}$) by institutional events. For example, the institutional event *give* could initiate the donor agent's permission to leave the stage, triggering the *absentation* event (equation 11). When the *interdiction* event is currently active and a *violation* event occurs, the interdiction event is terminated (20). Equations 15 to 21 in Figure 1 describe the initiation and termination of fluents in the Punch and Judy sausages scenario detailed in Section 2.1.

4 Regimenting agent actions with institutions

4.1 Institutions and multi-agent systems

Belief-Desire-Intention (BDI) agents' behaviour can be governed by running an institution manager in their environment, observing all agent actions and events. Given a set of observed events over time, such a manager can infer what permissions, obligations and institutitional powers hold at any given time.

The institution manager updates each agents' percepts to change their permissions and obligations. At each instant in time, the institution manager works out what an agent is permitted or obliged to do, then updates the agent's percepts (beliefs about the environment) with the set of permissions and obligations that hold at that time. It is up to the agent whether or not they act on these percepts.

As part of the BDI architecture of agents, an agent has beliefs about themselves, other agents and their environment. They also have goals that they desire to carry out (desires) and goals they intend to carry out next or are carrying out (intentions). The permissions and obligations that an agent receives from the institution manager only affect their beliefs: they believe that the norms of their world put certain expectations on them. These beliefs may or may not affect the plans that the agent desires or intends to carry out.

4.2 Describing institutions with InstAL and ASP

Answer Set Programming (ASP) [2] is a method of programming by specifying the requirements that a solution must fulfil. A specification of the constraints and rules of a problem are written and then queried, producing solutions in the form of answer sets.

Each line of an ASP program is a *rule*, which is a constraint that narrows down the set of solutions when queried. Rules consist of two parts: a head literal (l) and a body (B), separated with a left arrow: $l \leftarrow B$. If every literal in the body evaluates to *true*, then the head literal is also true.

Specifying our institution in ASP allows us to reason about the effects of events occuring over time. Given an institutional model and a sequence of events as input, the output would be the set of norms in the form of *permissions* and *obligations* that hold at certain instants in time.

To describe our institutional model, we use InstAL [4], a domain specific language for describing institutions that compiles to AnsProlog, a declarative programming language for Answer Set Programming (ASP) [2]. instAL's semantics are based upon the Situation Calculus [11] and the Event Calculus [6]. It is used to describe how external events generate institutional events, which can then initiate or terminate fluents that hold at certain instants in time. These fluents can include the permissions and obligations that describe what an agent is permitted or obligated to do at specific points in time.

Returning to the scenario in Section 2.1, if an agent with the role of *donor* leaves the stage, it generates the *absentation* Propp story function in the institution:

```
leaveStage(X) generates intAbsentation(X) if role(X, dispatcher),
    activeTrope(interdiction);
```

The *absentation* institutional event gives the crocodile permission to enter the stage if there are any sausages on the stage. It also terminates the permission of the absented agent to leave the stage, as they have already done so:

```
intAbsentation(X) initiates perm(enterStage(croc)) if objStage(sausages)
    ;
intAbsentation(X) terminates onStage(X), perm(leaveStage(X));
```

InstAL rules like those shown above are compiled into AnsProlog ASP rules describing which fluents hold at certain points in time. Once the InstAL model is compiled to AnsProlog, we use the *clingo* answer set solver [5] to ground the logical variables, and 'solve' queries by finding all permissions and obligations that apply to any agents, given a sequence of events as the query input. The agents' percepts are then updated with their permitted and obliged actions from that moment in time onwards.

Listing 1 shows how the sausages scenario would be described in ASP, for the first two events of the scene. Starting with an initial set of fluents that hold at $t0$, only fluents that have been initiated and not terminated hold at the next instant.

Listing 1 Sausages scenario in ASP

```
holdsat(perm(tellprotect(dispatcher, villain, item), t0).
occurred(tellprotect(dispatcher, villain, item), t0).
initiated(active(interdiction), t1).
initiated(perm(give(donor, villain, item)), t1).
terminated(tellprotect(dispatcher, villain, item), t1).
holdsat(perm(give(donor, villain, item)), t1).
holdsat(active(interdiction), t1).
occurred(give(donor, villain, item), t1).
initiated(active(receipt), t2).
initiated(perm(leavestage(donor)), t2).
terminated(perm(give(donor, villain, item)), t2).
holdsat(active(interdiction), t2).
holdsat(active(receipt), t2).
holdsat(perm(leavestage(donor)), t2).
```

4.3 Adding agent percepts from ASP solutions

With every event that occurs in the narrative, a query consisting of all events so far is sent to the solver. Its output tells us what permissions and obligations hold for certain agents at the next instant. These permissions and obligations are added to the agents' belief bases as percepts. The agents' plans are carried out based on these permissions and obligations.

For example, in the scene where Joey gives the sausages to Punch, Punch may see that he has permission to eat the sausages, drop them, fight the crocodile, run away (leave the stage) or shout for help at the crocodile or audience. His obligation for the scene, in accordance with the Punch and Judy narrative world, is to either eat the sausages himself, or let the crocodile sausages. This ends Propp's *interdiction* story function with a *violation* function. Note that his obligation is not to guard the sausages as asked to by Joey. While Joey's entrustment of the sausages is an obligation of sorts, Punch's only true obligations are to the narrative.

We have a prototype system where the agents choose their actions based on their emotional state. Before carrying out a potentially narrative-altering plan, each agent appeals to the audience for encouragement. They do this by turning to the audience and announcing their intentions. The audience then cheers or boos the character, which affects their emotional state, which is based on Russell's [14] circumplex model of emotion. In this model, a person's emotion is determined by three variables: Valence (positivity), Arousal and Dominance.

Depending on the action planned, a cheer or boo from the audience will raise or lower an agent's valence, arousal or dominance level. This changes the agents' motivation to select a certain permitted action to carry out as part of their plan.

In the above example, a depressed Punch may decide to violate his obligations by not eating the sausages and instead leave the stage with them. Alternatively, a furious Punch would viciously attack the crocodile, not allowing him to eat the sausages. This also violates the norms of the narrative world. However, for most emotional states the norms are observed by either Punch eating the sausages or letting the crocodile eat them.

5 Conclusion

With our approach to interactive narrative generation, we regiment the rules of the story domain using an institutional model. This model describes what each agent is permitted and obliged to do at any point in the story. Institutional regimentation of agents acting out a story using story-world norms allows much more flexibility than if the world's rules were strictly enforced. The deontic language of permissions and obligations allows the agents to act out small details of the narrative, while guiding them into an underlying narrative structure.

References

1 Vincent Baines and Julian Padget. A situational awareness approach to intelligent vehicle agents. In Michael Behrisch and Melanie Weber, editors, *Modeling Mobility with Open Data*, Lecture Notes in Mobility, pages 77–103. Springer International Publishing, 2015.

2 Chitta Baral. *Knowledge representation, reasoning and declarative problem solving*. Cambridge university press, 2003.

3 Pablo Cayetano Noriega Blanco-Vigil. *Agent mediated auctions: the fishmarket metaphor*. PhD thesis, Universitat Autònoma de Barcelona, 1998.

4 Owen Cliffe, Marina De Vos, and Julian Padget. Specifying and reasoning about multiple institutions. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pages 67–85. Springer, 2007.

5 Martin Gebser, Benjamin Kaufmann, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Marius Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.

6 Robert Kowalski and Marek Sergot. A logic-based calculus of events. In *Foundations of knowledge base management*, pages 23–55. Springer, 1989.

7 JeeHang Lee, Tingting Li, and Julian Padget. Towards polite virtual agents using social reasoning techniques. *Computer Animation and Virtual Worlds*, 24(3-4):335–343, 2013.

8 Seung Y Lee, Bradford W Mott, and James C Lester. Learning director agent strategies: An inductive framework for modeling director agents. In *Intelligent Narrative Technologies*, 2011.

9 Michael Mateas and Andrew Stern. Façade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference*, pages 4–8, 2003.

10 Vladimir Propp. *Morphology of the Folktale*. University of Texas Press, 2010.

11 Raymond Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. *Artificial intelligence and mathematical theory of computation: papers in honor of John McCarthy*, 27:359–380, 1991.

12 Justus Robertson and Robert Michael Young. Modelling character knowledge in plan-based interactive narrative to extend accomodative mediation. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

13 Juan A Rodriguez-Aguilar et al. *On the design and construction of Agent-mediated Institutions*. PhD thesis, Universidad Autónoma de Barcelona, 2001.

14 James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

15 Zach Tomaszewski. On the use of reincorporation in interactive drama. In *Intelligent Narrative Technologies*, 2011.

16 Javier Vázquez-Salceda. The role of norms and electronic institutions in multi-agent systems applied to complex domains. the harmonia framework. *AI Communications*, 16(3):209–212, 2003.

Good Timing for Computational Models of Narrative Discourse[*]

David R. Winer, Adam A. Amos-Binks, Camille Barot, and R. Michael Young

Department of Computer Science, North Carolina State University
Raleigh, NC, USA
{drwiner,aaamosbi,cbarot,rmyoung}@ncsu.edu

Abstract

The temporal order in which story events are presented in discourse can greatly impact how readers experience narrative; however, it remains unclear how narrative systems can leverage temporal order to affect comprehension and experience. We define structural properties of discourse which provide a basis for computational narratologists to reason about good timing, such as when readers learn about event relationships.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods, I.2.8 Problem Solving, Control Methods, and Search, I.2.7 Natural Language Processing

Keywords and phrases causal inference, narrative, discourse structure, computational model

Digital Object Identifier 10.4230/OASIcs.CMN.2015.152

1 Introduction

Narratologists frequently recognize that the temporal order in which story events are presented can greatly impact how readers comprehend narrative [6, 3, 1]. For example, readers usually notice when events are not presented in a possible storyworld chronology (e.g. flashbacks). Moreover, psychologists show that rearranging the order of events, while still presenting events in a possible storyworld chronology, affects how readers interpret narrative [13, 15, 14, 7]. Storytelling decisions about when readers should learn about event relationships have not received the same level of attention by narratologists compared to devices like flashback or flashforward. Computational narratologists interested in accounting for storytelling decisions about timing may benefit from encoding the relationship between temporal order of events in discourse presentation and comprehension in readers.

Our position is motivated by psychology research which demonstrates that rearranging events, while still presenting them in a possible storyworld chronology, affects how readers understand discourse. Consider an important event that has multiple relevant outcomes in a story. The order that readers learn about the outcomes can affect whether each outcome is interpreted as a direct result versus a side effect of the important event [13, 8]. Similarly, consider a situation where multiple antecedent events must occur for an outcome to occur. When readers think counterfactually about the outcome, research shows that readers are biased by temporal order when attributing causal responsibility to antecedent events and do not consider all antecedents equally [15, 9, 14, 7]. We believe these kinds of situations are

[*] This material is based upon work supported in whole or in part with funding from the Laboratory for Analytic Sciences (LAS). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 152–156
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

opportunities for storytellers to use good timing in nonlinear stories, but further evaluation is needed to predict more precisely how temporal order affects narrative experience.

Previous approaches for modeling narrative discourse presentation have not encoded in a general way how presentation ordering can affect inferences made by readers during comprehension. Computational models of reader comprehension used in narrative systems [10, 4, 11] simulate human reasoning to make decisions about narrative discourse presentation. These reader models are limited because they lack a simple underlying characterization of the ways that timing affects the reader's experience of the story. We believe that reader models can more accurately model narrative experiences like suspense and surprise by encoding the way reader comprehension is biased by temporal order.

In the work presented, we formally define structural properties of discourse which provide a basis for computational narratologists to reason about good timing in narrative discourse. This model clearly distinguishes the causal structure of story which drives comprehension [16, 5, 12] from the temporal properties of discourse. We believe that a formal approach that delineates causal structure from temporal discourse structure would greatly benefit experiment design investigating the role of timing on comprehension. If the effects of timing on comprehension were better understood, narrative analysis and generation systems could then account for good timing in an actionable way to interpret and produce interesting narrative experiences.

2 Story Structure

A conjunction of function-free ground literals is used to represent the state of the world, describing what is true and false in the story world. The initial state of the world contains the propositions that are initially true. Other states are established as the result of an event.

▶ **Definition 1** (Event). An *event* is a tuple $\langle P, E, V \rangle$ where P is a set of preconditions (literals that must be true before the event can be executed), E is a set of effects, literals made true by the event's execution, and V is a label which distinguishes the event.

▶ **Definition 2** (Causal Link). A *causal link* between two events s and t, denoted $s \xrightarrow{p} t$ indicates that s is an event which has effect p that enables a precondition p of event t. Event s is the *antecedent*, t is the *consequent*, and s and t are *causal partners*.

▶ **Definition 3** (Ordering Constraint). An *ordering constraint* of two events s and t denoted $s \prec t$ indicates that event s is *necessarily ordered* before event t.

Constraints are transitive: if $s \prec k$ and $k \prec t$, then $s \prec t$.

▶ **Definition 4** (Story Plan). A *story plan* Φ is a tuple $\langle S, O, L \rangle$ where S is a set of events, O is a set of ordering constraints over events in S, and L is a set of causal links over events in S.

A story plan is complete if and only if every precondition of every event is satisfied (by other events or by the initial state) and it is not possible that an event can occur between causal partners that reverses the effect of the antecedent enabling the consequent.

Figure 1 shows an example story plan which models a simplified sequence of events in the film *Indiana Jones and the Raiders of the Lost Ark*. Initially, Indiana Jones (IJ) and a Nazi (N) are fighting over a headpiece medallion (medal) which is embedded with the location of the Ark. During the fight, the medal is set on fire and becomes burning hot. The Nazi picks up the medal and his hand is burned, resulting in two outcomes. The first outcome is that

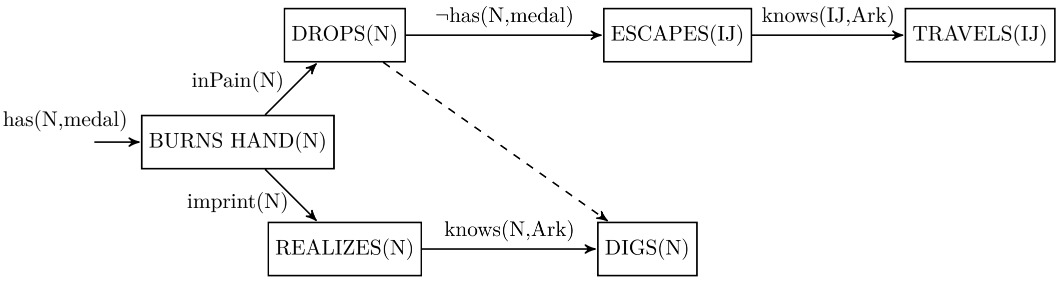


Figure 1 An example highly-simplified story plan of the Indiana Jones story. Ordering constraints are denoted by directed edges, with labeled edges indicating causal links.

the Nazi is in pain, causing him to drop the medal which enables Indiana Jones to escape with it and then travel to the Ark location. The second outcome is that the Nazi has the location from the medal imprinted into his hand. When he realizes this, he uses the location to choose a digging site.

3 Presentation Structure

The presentation of a story is a story plan where events are mapped to a total ordering in a sequential discourse structure.

▶ **Definition 5** (Presentation)**.** A *presentation* Ψ is a tuple $\langle \Phi, T \rangle$ where $\Phi = \langle S, O, L \rangle$ is a story plan and T is a bijection function $T : S \to [1, ..., n]$ with $n = |S|$ mapping events in S to a total ordering in $\mathbb{N}$.

A presentation $\langle \Phi, T \rangle$ is complete if and only if the story plan Φ is complete and if $\forall u, v \in S,\ u \prec v \in O \implies T(u) < T(v)$.

▶ **Definition 6** (Temporal Adjacency)**.** An event u is *temporally adjacent* to a causal partner v in a presentation Ψ if and only if $|T(u) - T(v)| = 1$.

▶ **Definition 7** (Intervening Discourse Event)**.** An event v is an *intervening discourse event (IDE)* for causal link $s \xrightarrow{p} t$ in a presentation $\Psi = \langle \Phi, T \rangle$ where $\Phi = \langle S, O, L \rangle$ if and only if $v, s, t \in S$, $s \xrightarrow{p} t \in L$, and $T(s) < T(v) < T(t)$.

▶ **Definition 8** (Temporal Separation)**.** An event u is *temporally separated* by separation size k from a causal partner v in a presentation $\Psi = \langle \langle S, O, L \rangle, T \rangle$ if and only if the number of IDEs for $u \xrightarrow{p} v$ is greater than k where $u, v \in S$ and $u \xrightarrow{p} v \in L$.

For simplicity, we do not encode differences between intervening discourse events such as the dimension of the situation [18, 2, 12], and therefore consider all events as equally weighted transitions of the world state.

In Figure 2, we show two presentations of the story plan from Figure 1. In Presentation A, a sequence resembling the order in the film, the events of Indiana Jones escaping with the medal (event 3) and traveling (event 4) are IDEs for causal link BURNS HAND$\xrightarrow{imprint}$ REALIZES. When these causal partners (events 1 and 5) are temporally separated, the consequent (event 5) may not be anticipated and perhaps will surprise the reader. However, in Presentation B, the same events BURNS HAND and REALIZES are temporally adjacent (events 1 and 2). This changes how the reader interprets the subsequent events, perhaps now anticipating that Indiana Jones will run into the Nazis at the Ark location.

The Indiana Jones Story
Init. The medallion is imprinted with the location of the Ark. The medallion is burning hot.
Presentation A. 1. The Nazi grabs the hot medallion and his hand is severely burned. 2. In pain, the Nazi drops the medallion. 3. Indiana Jones takes the medallion and escapes. 4. Indiana Jones travels to the destination indicated on the medallion. 5. The Nazi realizes the location from the medallion is imprinted onto his hand. 6. The Nazis dig for the Ark
Presentation B. 1. The Nazi grabs the hot medallion and his hand is severely burned. 2. The Nazi realizes the location is imprinted onto his hand. 3. In pain, the Nazi drops the medallion. 4. Indiana Jones takes the medallion and escapes. 5. Indiana Jones travels to the destination indicated on the medallion 6. The Nazi dig for the Ark.

Figure 2 Two presentations of the Indiana Jones story plan depicted in Figure 1.

The two presentations may elicit different narrative experiences because the temporal sequence affects the order that readers learn which events are *important*. A definition of causal importance, modeled as the number of incoming and outgoing causal connections of an event in a story plan, has proven effective at modeling human judgment [16, 17, 5, 4, 12]. Whenever a reader encounters a new event that has an antecedent in the story, the importance of that antecedent, from the reader's perspective, increases by virtue of the revealed causal connection. In the Indiana Jones Story, event 1 (BURNS) is the most important event in the story because it has two outgoing connections. In Presentation A, the reader does not learn of the event's importance until event 5, whereas in Presentation B, the event's importance is learned by event 3 which changes the context for interpreting the remaining events. In general, the timeline of when readers learn that events are more or less important may be a dimension of temporal discourse structure critical for characterizing narrative interpretation.

4 Summary

In the work presented, we provided a preliminary model with formally defined properties of story and discourse to act as a framework for reasoning about timing in narrative. One immediate application of our framework is that we can design experiments that tease out the effect of temporal order on comprehension and directly encode this with a computational model. This would enable generative systems to leverage timing in an actionable way for producing novel and more interesting experiences. Our framework currently captures only basic elements of story content and discourse timing to illustrate the relationship between causal structure and discourse presentation. The framework will be extended to identify relationships between discourse timing and other formally defined story content.

References

1 Edward Branigan. *Narrative comprehension and film.* Routledge, 1992.

2 Rogelio E. Cardona-Rivera, Bradley A. Cassell, Stephen G. Ware, and R. Michael Young. Indexter: A computational model of the event-indexing situation model for characterizing narratives. In *The Workshop on Computational Models of Narrative at the Language Resources and Evaluation Conference*, pages 32–41, 2012.

3 Seymour Benjamin Chatman. *Story and discourse: Narrative structure in fiction and film.* Cornell University Press, 1980.

4 Yun Gyung Cheong and R. Michael Young. Suspenser: A Story Generation System for Suspense. *IEEE Transactions on Computational Intelligence and AI in Games*, 11(4):1–1, 2014.

5 David B. Christian and R. Michael Young. Comparing cognitive and computational models of narrative structure. In *Proceedings of the 19th AAAI Conference on Artificial Intelligence*, pages 385–390, 2004.

6 Gérard Genette and Jane E. Lewin. *Narrative discourse: An essay in method.* Cornell University Press, 1983.

7 David A. Lagnado, Tobias Gerstenberg, and Ro'i Zultan. Causal responsibility and counterfactuals. *Cognitive science*, 37(6):1036–73, 2013.

8 John Mikhail. Universal moral grammar: Theory, evidence and the future. *Trends in cognitive sciences*, 11(4):143–152, 2007.

9 Jerome L. Myers, Makiko Shinjo, and Susan A. Duffy. Degree of causal relatedness and memory. *Journal of Memory and Language*, 26(4):453–465, 1987.

10 James. Niehaus and R. Michael Young. Cognitive models of discourse comprehension for narrative generation. *Literary and Linguistic Computing*, 29(4):561–582, 2014.

11 Brian O'Neill and Mark Riedl. Dramatis: A computational model of suspense. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 944–950, 2014.

12 Gabriel A. Radvansky, Andrea K. Tamplin, Joseph Armendarez, and Alexis N. Thompson. Different Kinds of Causality in Event Cognition. *Discourse Processes*, 51(7):601–618, 2014.

13 Stephen J. Read, Peter R. Druian, and Lynn Carol Miller. The role of causal sequence in the meaning of actions. *British journal of social psychology*, 28(4):341–351, 1989.

14 Susana Segura, Pablo Fernandez-Berrocal, and Ruth M. J. Byrne. Temporal and causal order effects in thinking about what might have been. *The Quarterly journal of experimental psychology. A, Human experimental psychology*, 55(4):1295–305, 2002.

15 Barbara A Spellman. Crediting causality. *Journal of Experimental Psychology: General*, 126(4):323–348, 1997.

16 Tom Trabasso and Linda L. Sperry. Causal relatedness and importance of story events. *Journal of Memory and language*, 24(5):595–611, 1985.

17 R. Michael Young. Using grice's maxim of quantity to select the content of plan descriptions. *Artificial Intelligence*, 115(2):215–256, 1999.

18 Rolf A. Zwaan and Gabriel A. Radvansky. Situation models in language comprehension and memory. *Psychological bulletin*, 123(2):162, 1998.

Model-based Story Summary

Patrick Henry Winston

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA, USA
phw@mit.edu

Abstract

A story summarizer benefits greatly from a reader model because a reader model enables the story summarizer to focus on delivering useful knowledge in minimal time with minimal effort. Such a summarizer can, in particular, eliminate disconnected story elements, deliver only story elements connected to conceptual content, focus on particular concepts of interest, such as revenge, and make use of our human tendency to see causal connection in adjacent sentences. Experiments with a summarizer, built on the Genesis story understanding system, demonstrate considerable compression of an 85-element précis of the plot of Shakespeare's *Macbeth*, reducing it, for example, to the 14 elements that make it a concise summary about Pyrrhic victory. Refocusing the summarizer on regicide reduces the element count to 7, or 8% of the original.

1998 ACM Subject Classification I.2.0 General/Cognitive simulation

Keywords and phrases story telling and summarization, story understanding, cognitive modeling

Digital Object Identifier 10.4230/OASIcs.CMN.2015.157

1 Vision

Suppose you want a program to summarize a story. How should your program decide what to include and what to leave out? I suggest that people read summaries mainly to acquire useful knowledge in minimal time with minimal effort. Thus, a summary program should focus on knowledge useful as precedent, exclude obvious inferences, but include reflective inferences that help the reader understand how the key elements are connected. Accordingly, a summary program should adhere to several principles reminiscent of the maxims of Grice [5], and in so adhering, a summary program must have an understanding of human story understanding in general and of the summary reader in particular. My students and I have built such an understanding into our Genesis story-understanding system, and we can adjust Genesis to model the knowledge and interests of particular summary readers.

2 Genesis models aspects of story understanding by humans

Much recent work has focused on applications that digest large amounts of data so as to exhibit a kind of intelligence. Google's caption generator [14], for example, is no doubt an engineering marvel, but it sheds little or no light on our human visual faculty. Likewise, IBM's Watson [1] is no doubt intelligent in some ways, but it does not think as we think.

Work on Genesis goes in a different direction. Genesis was developed in the belief that story understanding and telling is the distinguishing feature of human intelligence [15, 16, 17]. The aim in building Genesis is to model aspects of that story understanding and telling feature at the expense of working with story summaries written in simple English of the kind we can get through the START parser [6] and into Genesis's inner language of relations and events.

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 157–165
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One such simple Genesis-readable story is the following précis, which is based loosely on Shakespeare's play, *Macbeth*. It is itself a summary, but it is also an anvil on which to hammer out principles that enable further compression and clarification.

> **Macbeth précis**
>
> Scotland and England are countries. Dunsinane is a castle and Birnam Wood is a forest. Macbeth, Macduff, Malcolm, Donalbain, Lady Macbeth, Lady Macduff, Cawdor, and Duncan are persons. Lady Macbeth is Macbeth's wife. Lady Macduff is Macduff's wife. Lady Macbeth is evil and greedy. Duncan is the king, and Macbeth is Duncan's successor. Duncan is an enemy of Cawdor. Macbeth is brave. Macbeth defeats Cawdor. Duncan becomes happy because Macbeth defeats Cawdor. The witches are weird. The witches meet at night. The witches danced and chanted. Macbeth tells witches to speak. Macbeth talks with the witches. Birnam Wood is a forest. Witches predict that Birnam Wood will go to Dunsinane. The witches predict that Macbeth will become Thane of Cawdor. The witches predict that Macbeth will become king. The witches astonish Macbeth. Duncan executes Cawdor. Macbeth becomes Thane of Cawdor. Duncan rewarded Macbeth because Duncan became happy. Lady Macbeth wants Macbeth to become king. Macbeth is weak and vulnerable. Lady Macbeth persuades Macbeth to want to become the king because Lady Macbeth is greedy. Macbeth loves Lady Macbeth. Macbeth wants to please lady Macbeth. Macbeth wants to become king because Lady Macbeth persuaded Macbeth to want to become the king. Lady Macbeth plots to murder the king with Macbeth. Macbeth invites Duncan to dinner. Duncan compliments Macbeth. Duncan goes to bed. Duncan's guards become drunk and sleep. In order to murder Duncan, Macbeth murders the guards, Macbeth enters the king's bedroom, and Macbeth stabs Duncan. Macbeth becomes king. Malcolm and Donalbain become afraid. Malcolm and Donalbain flee. Macbeth's murdering Duncan leads to Macduff's fleeing to England. In order to flee to England, Macduff rides to the coast and Macduff sails on a ship. Macduff's fleeing to England leads to Macbeth's murdering Lady Macduff. Macbeth hallucinates at a dinner. Lady Macbeth says he hallucinates often. Everyone leaves because Lady Macbeth tells everyone to leave. Macbeth's murdering Duncan leads to Lady Macbeth's becoming distraught. Lady Macbeth has bad dreams. Lady Macbeth thinks she has blood on her hands. Lady Macbeth tries to wash her hands. Lady Macbeth kills herself. Birnam Wood goes to Dunsinane. Macduff's army attacks Dunsinane. Macduff curses Macbeth. Macbeth refuses to surrender. Macduff kills Macbeth.

Given the *Macbeth* précis, Genesis notes and infers several kinds of causal connections. Connections noted are those signaled by the word *because*, the words *leads to*, and the words *in order to* in stories. *Because* signals a direct cause between story elements (Duncan becomes happy because Macbeth defeated Cawdor); *leads to* indicates there is a chain of unstated causes connecting two story elements (Macbeth's murdering Duncan leads to Macduff's fleeing to England); *in order to* explains how something is done (In order to murder Duncan, Macbeth murders the guards, Macbeth enters the king's bedroom, and Macbeth stabs Duncan).

2.1 Genesis deploys various kinds of common-sense rules

In addition to noting explicit causal connections, Genesis produces other causal connections using inference rules, including deduction rules, abduction rules, explanation rules, and

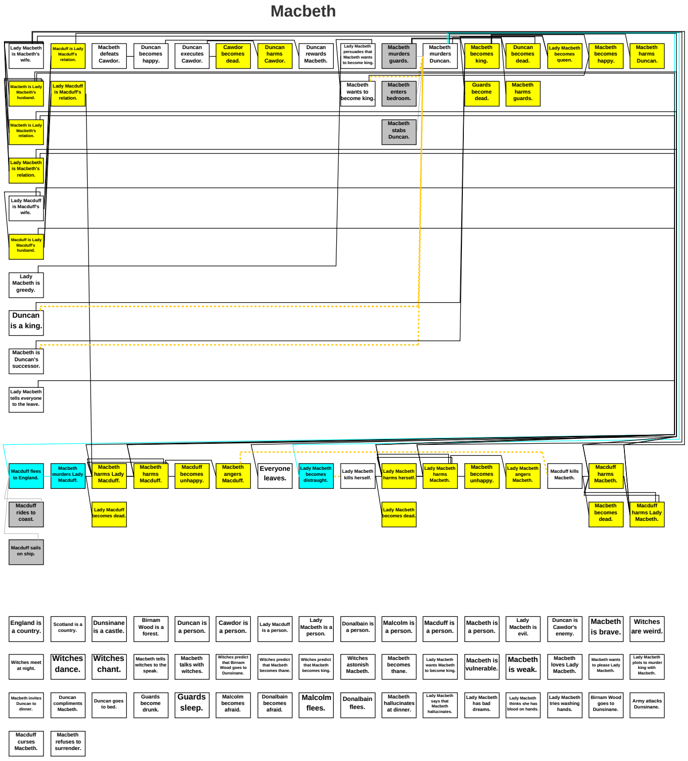

Macbeth
Lady Macbeth is Macbeth's wife.
Macduff is Lady Macduff's relation.
Macbeth defeats Cawdor.
Duncan becomes happy.
Duncan executes Cawdor.
Cawdor becomes dead.
Duncan harms Cawdor.
Duncan rewards Macbeth.
Lady Macbeth persuades that Macbeth wants to become king.
Macbeth murders guards.
Macbeth murders Duncan.
Macbeth becomes king.
Duncan becomes dead.
Lady Macbeth becomes queen.
Macbeth becomes happy.
Macbeth harms Duncan.
Macbeth is Lady Macbeth's husband.
Lady Macduff is Macduff's relation.
Macbeth wants to become king.
Macbeth enters bedroom.
Guards become dead.
Macbeth harms guards.
Macbeth is Lady Macbeth's relation.
Macbeth stabs Duncan.
Lady Macbeth is Macbeth's relation.
Lady Macduff is Macduff's wife.
Macduff is Lady Macduff's husband.
Lady Macbeth is greedy.
Duncan is a king.
Macbeth is Duncan's successor.
Lady Macbeth tells everyone to the leave.
Macduff flees to England.
Macbeth murders Lady Macduff.
Macbeth harms Lady Macduff.
Macbeth harms Macduff.
Macduff becomes unhappy.
Macbeth angers Macduff.
Everyone leaves.
Lady Macbeth becomes distraught.
Lady Macbeth kills herself.
Lady Macbeth harms herself.
Lady Macbeth harms Macbeth.
Macbeth becomes unhappy.
Lady Macbeth angers Macbeth.
Macduff kills Macbeth.
Macduff harms Macbeth.
Macduff rides to coast.
Lady Macduff becomes dead.
Lady Macbeth becomes dead.
Macbeth becomes dead.
Macduff harms Lady Macbeth.
Macduff sails on ship.
England is a country.
Scotland is a country.
Dunsinane is a castle.
Birnam Wood is a forest.
Duncan is a person.
Cawdor is a person.
Lady Macduff is a person.
Lady Macbeth is a person.
Donalbain is a person.
Malcolm is a person.
Macduff is a person.
Macbeth is a person.
Lady Macbeth is evil.
Duncan is Cawdor's enemy.
Macbeth is brave.
Witches are weird.
Witches meet at night.
Witches dance.
Witches chant.
Macbeth tells witches to the speak.
Macbeth talks with witches.
Witches predict that Birnam Wood goes to Dunsinane.
Witches predict that Macbeth becomes thane.
Witches predict that Macbeth becomes king.
Witches astonish Macbeth.
Macbeth becomes thane.
Lady Macbeth wants Macbeth to become king.
Macbeth is vulnerable.
Macbeth is weak.
Macbeth loves Lady Macbeth.
Macbeth wants to please Lady Macbeth.
Lady Macbeth plots to murder king with Macbeth.
Macbeth invites Duncan to dinner.
Duncan compliments Macbeth.
Duncan goes to bed.
Guards become drunk.
Guards sleep.
Malcolm becomes afraid.
Donalbain becomes afraid.
Malcolm flees.
Donalbain flees.
Macbeth hallucinates at dinner.
Lady Macbeth says that Macbeth hallucinates.
Lady Macbeth has bad dreams.
Lady Macbeth thinks she has blood on hands.
Lady Macbeth tries washing hands.
Birnam Wood goes to Dunsinane.
Army attacks Dunsinane.
Macduff curses Macbeth.
Macbeth refuses to surrender.

Figure 1 Elaboration graph generated by the *Macbeth* précis. Connections are color coded: deduction rules and explicit *because* connections produce black lines; explicit *leads to* connections produce blue lines; explanation rules produce orange connections. You can expand the diagram if you are using a PDF viewer.

presumption rules. Deduction rules, such as *If x kills y, then y becomes dead*, make connections whenever all their antecedents are in a story. Abduction rules make connections between elements and presumed antecedents. For example, Genesis's reader model may include the abduction rule *If x kills y, then x must be insane.* Explanation rules make connections only when there is no other known way to explain an element. For example, *Macduff kills Macbeth* is explained by the explanation rule *If x angers y, then y may kill x* and the previously inferred element *Macbeth angers Macduff.* Presumption rules, like abduction rules, make connections between elements and presumed antecedents, but only when there is no other known way to explain an element. Presumption rules, unlike explanation rules, do not require antecedents to be already in place. Abduction rules, explanation rules, and presumption rules are ranked, so that the highest ranking rule dominates in the event multiple rules are available for explaining an unexplained event. We intend to develop a more sophisticated, context-sensitive process.

The noted and inferred causal connections constitute the *elaboration graph* of causally connected elements as shown in Figure 1.

2.2 Genesis discovers concepts by searching for connections

Genesis finds concepts in the elaboration graph by searching for elements that instantiate *concept patterns.* In general, concept patterns include specifications for sequences of causal relations that start and end with particular, specified elements. The concept pattern for

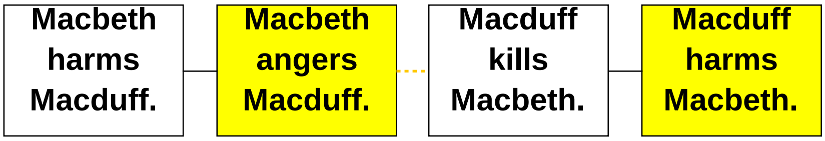


Figure 2 The instantiated revenge concept pattern found in the *Macbeth* précis.

revenge, for example, is just a single such sequence described by *x's harming y leads to y's harming x*. An instantiated revenge pattern is shown in Figure 2.

Remarkably, the elaboration graph, augmented by discovered concept patterns, provides the substrate for developing models of many kinds of story understanding and telling, including question answering, cultural bias in interpretation, instructional telling with a learner model, persuasive telling with a listener model, precedent-based prediction, and as described here, summary.

2.3 We provide common-sense rules and concept patterns in English

My students and I provide Genesis with common-sense rules, concept patterns, and stories; all rules, patterns, and stories are provided in English as indicated in the examples. Our purpose is to establish, by telling, what Genesis needs to know to exhibit a kind of humanlike understanding.

We think it reasonable, at this stage, to tell Genesis what it needs to know. One reason is that much of what we know we learn by being told. Few would have the concept of Pyrrhic victory, for example, without being told. Another reason is that much of what we tell Genesis in experimenting with one story finds use in other stories. Revenge, for example, is revenge not only in *Macbeth*, but also in fairy tales and international conflicts. Yet another reason is that we have done research on learning concept patterns from ensembles of stories [2, 7], and we are engaged in research on learning common sense by mining various textual sources.

3 The Genesis model enables principle-based story summary

Genesis, as a model of story understanding by humans, suggests several principles for summary. Some compress the story provided; others expand the story by adding helpful explanations. All work toward helping the reader to focus on the elements that convey useful knowledge and to grasp how the useful story elements are connected.

In the following, I articulate several such principles, and I explain how those principles are reflected in a model of story summarization by humans. I also show how the Genesis story summarizer, based on that model, performs on a test case.

3.1 The principle of connection

Good precedents exhibit causal connections between events that are likely to be seen again in future situations, thereby enabling understanding, prediction, and control. Accordingly, the Genesis story summarizer preserves those explicit story elements that are involved in causal connections, where the causal connections are either explicit or inferred. Genesis filters out explicit story elements that are neither an antecedent nor a consequent in any kind of causal connection.

The *Macbeth* précis contains 55 sentences, which, when understood by Genesis, expand to 85 explicit story elements, with the expansion caused by separately counting elements that are embedded in compound sentences and explicit causal connections and by adding one to the element count for each explicit causal connection. In what follows, I compare the number of summary elements with the number of explicit story elements for various versions of the Genesis summarizer.

Many of the explicit elements are not involved in causal connections of any kind, explicit or inferred, and thus offer little or nothing by way of constraining precedent. Keeping only those explicit elements that are causal connections and explicit elements that are embedded in Genesis's inferred causal connections produces the following summary in which the START system produces the English, with occasional awkwardness, from Genesis's inner language of relations and events:

Macbeth, with principle of connection

> Lady Macbeth is Macbeth's wife. Lady Macduff is Macduff's wife. Duncan is a king. Macbeth is Duncan's successor. Duncan becomes happy because Macbeth defeats Cawdor. Duncan executes Cawdor. Duncan rewards Macbeth because Duncan becomes happy. Lady Macbeth persuades that Macbeth wants to become king because Lady Macbeth is greedy. Macbeth wants to become king because Lady Macbeth persuades that Macbeth wants to become king. In order to murder Duncan, Macbeth murders guards; in order to murder Duncan, he enters bedroom; in order to murder Duncan, he stabs Duncan. Donalbain is Duncan's son. Malcolm is Duncan's son. For Macbeth to murder Duncan leads to Macduff's fleeing to England. In order to flee to England, Macduff rides to coast; in order to flee to it, he sails on ship. For Macduff to flee to England leads to Macbeth's murdering Lady Macduff. Everyone leaves because Lady Macbeth tells everyone to the leave. For Macbeth to murder Duncan leads to Lady Macbeth's becoming distraught. Lady Macbeth kills herself. Macduff kills Macbeth.

Thus, the *principle of connection* allows the Genesis summarizer to reduce the number of summary elements to 34, 40% of the 85 explicit story elements.

3.2 The principle of concept focus

Good precedents tend to be told in a manner that focuses attention on conceptual content because associating a story with its conceptual content is part of what separates novices from domain experts [3, 4]. Accordingly, another version of the Genesis story summarizer includes only explicit elements that lead eventually—via a chain of inferred connections—to an element lying in an instantiated concept pattern.

The elaboration graph plays a central role in this kind of summary because searches in the elaboration graph discover concepts and because searches in the elaboration graph determine which explicit elements are connected to those concepts. Filtering out other elements produces the following *Macbeth* summary:

Macbeth, with principle of concept focus added

> The story is about Regicide, Mistake because unhappy, Answered prayer, Revenge, Suicide, Mistake because harmed, Success, and Pyrrhic victory. Lady Macbeth is Macbeth's wife. Lady Macduff is Macduff's wife. Lady Macbeth persuades that Macbeth wants to become king because Lady Macbeth is greedy. Macbeth wants to become king because Lady Macbeth persuades that Macbeth wants to become king.

> In order to murder Duncan, Macbeth murders guards; in order to murder Duncan, he enters bedroom; in order to murder Duncan, he stabs Duncan. Macbeth murders Duncan, probably because Macbeth wants to become king, Duncan is a king, and Macbeth is Duncan's successor. For Macbeth to murder Duncan leads to Macduff's fleeing to England. In order to flee to England, Macduff rides to coast; in order to flee to it, he sails on ship. For Macduff to flee to England leads to Macbeth's murdering Lady Macduff. For Macbeth to murder Duncan leads to Lady Macbeth's becoming distraught. Lady Macbeth kills herself, probably because Lady Macbeth becomes distraught. Macbeth becomes unhappy. Macduff kills Macbeth, probably because Macbeth angers Macduff.

Now the summary contains only 30 of the 85 explicit story elements or 35%. Excluded are elements such as *Duncan becomes happy because Macbeth succeeded*, and *Duncan rewarded Macbeth because Duncan becomes happy.* None of the elements involved leads to an element in an instantiated concept.

3.3 The principle of dominant concept focus

Good precedents tend to have a particular purpose and focus attention on one or a few key concepts. Accordingly, yet another version of the Genesis story understander retains an explicit story element only if that element is connected via a chain of inferences to a key concept.

Which of the discovered concepts are the key concepts? There are several reasonable possibilities with which we propose to experiment once we have a large enough corpus of Genesis-readable stories, including concepts that cover a lot of the elements of the story over a long time span, concepts that involve violent acts, such as murder, concepts that excite big emotional reaction, concepts that indicate a dramatic situation, such as those identified by Polti, concepts that the summarizer wants the reader to note, concepts that the summarizer knows the reader wants to note, concepts that are rarely observed, and concepts that involve memorable elements.

For example, in the *Macbeth* précis, Pyrrhic victory dominates all other concepts in the sense that it incorporates the most story elements. Using Pyrrhic victory to summarize, rather than all concepts, Genesis produces the following:

> **Macbeth, with principle of dominant concept focus added**
>
> The story is about Pyrrhic victory. Lady Macbeth is Macbeth's wife. Lady Macduff is Macduff's wife. Lady Macbeth persuades that Macbeth wants to become king because Lady Macbeth is greedy. Macbeth wants to become king because Lady Macbeth persuades that Macbeth wants to become king. In order to murder Duncan, Macbeth murders guards; in order to murder Duncan, he enters bedroom; in order to murder Duncan, he stabs Duncan. Macbeth murders Duncan, probably because Macbeth wants to become king, Duncan is a king, and Macbeth is Duncan's successor. For Macbeth to murder Duncan leads to Macduff's fleeing to England. In order to flee to England, Macduff rides to coast; in order to flee to it, he sails on ship. For Macduff to flee to England leads to Macbeth's murdering Lady Macduff. Macduff kills Macbeth, probably because Macbeth angers Macduff.

The elements that deal with Lady Macbeth's suicide drop out; the number of summary elements is 25, 29% of the explicit story elements.

Memorable elements, incidentally, are readily captured in simple concept patterns that may involve no *leads to* elements, such as this *Memorable event* pattern: *a woman becomes the bishop.* Of course, what constitutes a memorable event may not be so memorable at a different time or place.

3.4 The principle of interpretation transparency

Good summaries do not require readers to guess how the summarizer has reasoned. Accordingly, the Genesis story summarizer is explicit about the assumptions it makes. In particular, the Genesis story summarizer includes not only the consequents of explanation rules, which are explicit in the story, but also the fully instantiated explanation rule, even though the antecedents themselves may be the consequents of deduction rules and not ordinarily included.

For example, the previous two summaries include *Macduff kills Macbeth, probably because Macbeth angers Macduff.* The rationale is that the summarizer, in eagerness to create a more coherent and easily understood story, has added something not completely obvious about how the summarizer has interpreted the story. Thus the summarizer's reasoning is transparent and the reader is relieved of reasoning effort.

3.5 Compression by eliminating details of how actions are performed

Good summaries stick to essentials. Accordingly, the Genesis story summarizer can be directed to eliminate details of how actions are performed, providing further compression.

Impatient readers will not care, for example, about exactly how Macbeth murders Duncan, so the Genesis story summarizer suppresses details about the guards, the bedroom, and stabbing:

Macbeth, with detail suppression added

> The story is about Pyrrhic victory. Lady Macbeth is Macbeth's wife. Lady Macduff is Macduff's wife. Lady Macbeth persuades that Macbeth wants to become king because Lady Macbeth is greedy. Macbeth wants to become king because Lady Macbeth persuades that Macbeth wants to become king. Macbeth murders Duncan, probably because Macbeth wants to become king, Duncan is a king, and Macbeth is Duncan's successor. For Macbeth to murder Duncan leads to Macduff's fleeing to England. For Macduff to flee to England leads to Macbeth's murdering Lady Macduff. Macduff kills Macbeth, probably because Macbeth angers Macduff.

With means deleted, the number of summary elements is further reduced to 18, 21% of the explicit story elements.

3.6 Compression using the post hoc ergo propter hoc assumption

Good summaries refrain from making natural inferences explicit because making them explicit is unnatural and annoying. Accordingly, the Genesis story summarizer supposes the reader will instinctively find plausible causal connections between adjacent events.

After this does not mean *because of this* in logic, but we use it nevertheless in telling stories smoothly, dropping explicit cause when proximity makes the cause apparent:

Macbeth, with post hoc ergo propter hoc processing added

> The story is about Pyrrhic victory. Lady Macbeth is Macbeth's wife. Lady Macduff is Macduff's wife. Lady Macbeth persuades that Macbeth wants to become king because

> Lady Macbeth is greedy. Macbeth wants to become king. Macbeth murders Duncan, probably because Duncan is a king, and Macbeth is Duncan's successor. Macduff flees to England. Macbeth murders Lady Macduff. Macduff kills Macbeth, probably because Macbeth angers Macduff. Macduff.

Processing with post hoc ergo propter hoc transforms *Macduff's fleeing to England leads to Macbeth murders Lady Macduff* to *Macbeth murders Lady Macduff.* With post hoc ergo propter hoc in play, the number of summary elements is 15, 18% of the explicit story elements.

4 Experiments

Using Genesis to summarize Shakespearian play summaries and cyberwar summaries produced the following percentages of summary elements relative to total elements. The Connected column reports the fraction of the explicit story elements that are reported when reporting all and only the elements in the story that are causally connected; the All-methods column reports the fraction of the explicit story elements reported when all of the principles here described are engaged.

	Connected	**All methods**
Macbeth	40%	18%
Hamlet	41%	14%
Estonia vs. Russia	40%	60%
Georgia vs. Russia	26%	19%

The compression numbers are not dramatic because the test stories are already summaries. The numbers generally drop when limiting the summary to elements that lead eventually to one or more instantiated concept patterns. One exception is *Estonia vs. Russia.* In this summary, one concept pattern is *Aggression of a bully*, a concept pattern that looks for which side the reader is friendly with: *x is my friend. x's angering y leads to y's harming x.* Instantiating that concept pattern brings in *I am Estonia's friend*, a disconnected element, but an element that corresponds to an element in the concept pattern. If the reader happens to be Russia's friend, the concept pattern triggered is *Teaching a lesson* and *I am Russia's friend* is included.

5 Contributions

Work on the Genesis story understanding and telling system has been inspired, in part, by the pioneering work of Roger Shank and his students [8, 9, 10, 11]. Work on Genesis has also been inspired, in part, by paleoanthropologist Ian Tattersall's reflections on what makes us human [12, 13], which led me to the conclusion that story understanding and story telling plays a major role. I have focused here on principles of story summary and shown how those principles are reflected the Genesis story summarizer. In particular, I have:

- Argued that a reader model is a necessary foundation for good story summary
- Identified the principles of connection, concept focus, dominant concept focus, and interpretation transparency.
- Suggested means compression and introduced post hoc ergo propter hoc processing.
- Exhibited an implemented, principle-based summarizer at work on a representative story from the Genesis library, a précis of *Macbeth*, showing a compression of 84%.

References

1 David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. The AI behind watson—the technical article. *AI Magazine*, 2010.

2 Mark A. Finlayson. *Learning Narrative Structure from Annotated Folktales*. PhD thesis, Electrical Engineering and Computer Science Department, MIT, Cambridge, MA, 2012.

3 Mark A. Finlayson and Patrick Henry Winston. Intermediate features and informational-level constraint on analogical retrieval. In *Proceedings of the 27th Annual Meeting of the Cognitive Science Society*, pages 666–671, 2005.

4 Dedre Gentner and Arthur B Markman. Structure mapping in analogy and similarity. *American Psychologist*, 52(1):45–56, 1997.

5 H. Paul Grice. *Studies in the Way of Words*. Harvard University Press, Cambridge, MA, 1989.

6 Boris Katz, Gary Borchardt, and Sue Felshin. Syntactic and semantic decomposition strategies for question answering from multiple resources. In *Proceedings of the AAAI 2005 Workshop on Inference for Textual Question Answering*, 2005.

7 Caryn Krakauer and Patrick Henry Winston. Story retrieval and comparison using concept patterns. In Mark Alan Finlayson, Pablo Gervas, Deniz Yuret, and Floris Bex, editors, *Proceedings of the 3rd Workshop on Computational Models of Narrative (CMN'12)*, volume 3, pages 119–124. European Language Resources Association (ELRA), 2012.

8 Wendy Lehnert. Plot units and narrative summarization. *Cognitive Science*, 5(4):293–331, 1981.

9 Roger C. Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3(4):552–631, 1972.

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

11 Roger C. Schank and Christopher K. Riesbeck. *Inside Computer Understanding: Five Programs Plus Miniatures*. Lawrence Erlbaum Associates, Mahwah, NJ, 1981.

12 Ian Tattersall. *Becoming Human*. Harcourt, Orlando, FL, 1998.

13 Ian Tattersall. Human evolution and cognition. *Theory in Biosciences*, 123(4):371–379, 2010.

14 Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *Cornell University Library digital repository, arXiv:1411.4555 [cs.CV]*, 2014.

15 Patrick Henry Winston. The strong story hypothesis and the directed perception hypothesis. In Pat Langley, editor, *Technical Report FS-11-01, Papers from the AAAI Fall Symposium*, pages 345–352, Menlo Park, CA, 2011. AAAI Press.

16 Patrick Henry Winston. The next 50 years: a personal view. *Biologically Inspired Cognitive Architectures*, 1:92–99, 2012.

17 Patrick Henry Winston. The right way. *Advances in Cognitive Systems*, 1:23–36, 2012.