Comparative Evaluation of Elementary Plot Generation Procedures

Pablo Gervás

Instituto de Tecnología del Conocimiento - Facultad de Informática,
Universidad Complutense de Madrid
Ciudad Universitaria, 28040 Madrid, Spain
pgervas@ucm.es
WWW home page: http://nil.fdi.ucm.es/

Abstract. There are many different abstractions as to what a 'story-telling' mechanism might be, each based on a particular understanding of what makes stories come together as a whole. Examples may be: archetypical instances of plot, story grammars, or the famous canonical sequence of character functions proposed by Vladimir Propp. From a computational point of view, each of these mechanisms can be used to construct or to validate new stories. The present paper carries out a comparative evaluation of a number of plot generation procedures, by grounding all of them on a basic reference vocabulary for the representation of narrative units, and applying to all of them a set of metrics distilled from the same procedures. The resulting set of computational tools is used in combination for comparative evaluation.

Keywords: computational creativity, narrative, story grammars, character functions, metrics for narrative

1 Introduction

Humans have for a long time tried to understand how it is that they can come up with stories that make sense and are enjoyable. In this endeavour, many different abstractions as to what a 'story-telling' mechanism might be have arisen. Each of these mechanisms is based on a particular understanding of what makes stories come together as a whole. Examples of how these different understandings of the essence of storyness are captured may be: archetypical instances of plot, story grammars, or the canonical sequence of character functions proposed by Vladimir Propp. From a computational point of view, each of these ways of capturing what makes a story work can be understood in two different ways: either as a procedure for constructing new stories or as a procedure for determining whether a given candidate sample is a valid story. Each of these mechanisms must be considered at most one possible simplification of the much more complex problem which is story-telling.

The present paper carries out a comparative evaluation of a number of plot generation procedures, by grounding all of them on a basic reference vocabulary for the representation of narrative units, and applying to all of them a set of metrics distilled from the same procedures.

2 Previous Work

For the purposes of this paper we focus on an abstract view of stories that concentrates on their overall narrative structure without considering details beyond a particular level of abstraction. The level of abstraction we select is that of large-grain description of activities of character that are relevant to the narrative structure. We refer to this abstract view of a story as *plot*.

Work on modelling the abstract structure of story has taken place both at the theoretical level – models of story structure in abstract terms – and at the computational level – computational implementations for story construction.

2.1 Theoretical Abstractions of Story Structure

Vladimir Propp [9] identified a set of regularities in a subset of the corpus of Russian folk tales and formulated them 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 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 and the relative order of appearance of these functions was noticeably stable. This led him to postulate that all these tales could be considered instances of a single structure.

Gervás et al [6] reviewed existing work on the description of plot to propose a set of schemas, compiled from various sources, and expressed in terms of sequences of character functions. The character functions employed came from an extended set, which used Propp's basic 31 as a seed and added additional elements were necessary to cover the features expressed in the reviewed descriptions of plot.

Gervás et al [5] further extended this work and devised an extended set of units of abstraction for narrative, equivalent to character functions, but defined as a vocabulary for the annotation of a corpus of plots of musicals. [1]

George Lakoff attempted [7] a reformulation of Propp's account of Russian folk tales as a transformational grammar in Chomsky's style,[2] then very much

[1] Because the vocabulary was developed to help a large set of volunteer annotators, and the term "character function" was considered confusing, and the term *plot element* was used instead.

[2] A number of grammar-based descriptions of story structure are reviewed in this paper. Each of them was originally formulated with a different notation. To make it easier to understand the differences and similarities across the different solutions reviewed, an attempt has been made to unify then into a single notation. This notation includes elements required to represent the peculiarities appearing over the complete set, but does not match a particular formalism. In the rules presented below, elements appearing in a sequence would appear in the same order in the

in vogue. The paper argues for the potential of different formal mechanisms to capture different aspects of the complexity of stories as identified in Propp's account, but the grammar described is actually incomplete (with rules missing for certain non-terminal symbols used). A simplified version[3] is provided in Table 1.

Plot → ComplicatingSequence ResolvingSequence
ResolvingSequence → (Episode) Resolution `trigger_resolved` Reward
ResolvingSequence → DonorSequence (Episode) Resolution `trigger_resolved` Reward
DonorSequence → `test_by_donor` `hero_reaction` `acquisition_magical_agent` **`use_magical_agent`**
Resolution → `struggle` `victory` | `difficult_task` `task_resolved`
ComplicatingSequence → (HelplessnessSequence) Complication `begin_counteraction`
Complication → `villainy` | `lack`
HelplessnessSequence → `interdiction` Violation
Violation → WilfullViolation | DeceptionByVillain SubmissionOfHero

Table 1. Lakoff's reinterpretation of Propp's morphology for Russian Folk Tales as a grammar

Rumelhart [10] pioneered the study of the structure of stories in the form of a grammar. Rumelhart suggests that the grammar he developed "accounts in a reasonable way for the structure of a wide range of simple stories". Rumelhart's grammar includes a set of syntactical rules that generate the constituent structure of stories and a parallel set of semantic interpretation rules which "determine the semantic representation of the story". This aspect of Rumelhart's work has received less attention than the syntactical rules. Rumelhart's syntactic rules, and the associated semantic interpretation rules, are presented in Table 2.

Thorndyke [11] carried out a set of experiments on the comprehension and recall of narrative discourse, and used for this a simplified version of Rumelhart's grammar. A simple transcription of this grammar is given in Table 3.

Computational Approaches to Story Generation Although story grammars were discredited as an actual model of human cognitive processing of stories [1], they remained a popular technique with researchers in story generation. The Joseph system [8] and the BRUTUS system [2] were based on story grammars. They both produced a succesful number of stories of high quality.

The Propper system [3] is a computational implementation of the procedure for generating stories described by Propp [9]. It uses Propp's canonical sequence of character functions for Russian folk tales, selects character functions out of it at random and places them in the same relative order in an output sequence. The revised version presented in [4] describes extensions to the original constructive

presentation of a story, → is used to indicate that the single term to the left can be rewritten (built) as the sequence of terms to the right, | indicates disjunction (choice), simple brackets are used to indicate optional elements, and square brackets are used to indicate that one or many of the corresponding elements should be included.

3 References to Proppian character functions are all transcribed in terms of a unified vocabulary (as defined in [3, 4]) and represented in `typewriter` font.

Story → Setting Episode	ALLOW(Setting,Episode)
Setting → [State]	AND(States)
Episode → [Event] Reaction	(ALLOW(Event,Event) | CAUSE(Event,Event)), INITIATE(Event,Reaction)
Event → Change-of-state	
Event → Action	
Event → Episode	
Reaction → InternalResponse OvertResponse	MOTIVATE(InternalResponse,OvertResponse)
InternalResponse → Emotion | Desire	
OvertResponse → Action | *Attempt	
Attempt → Plan Application	MOTIVATE(Plan,Application)
Application → (*Preaction) Action Consequence	CAUSE(Action,Consequence) | INITIATE(Action,Consequence) | ALLOW(Action,Consequence)
Preaction → Subgoal Attempt	MOTIVATE(Subgoal,Attempt)
Consequence → Reaction | Event	

Table 2. Rumelhart's syntactical and semantic interpretation rules

Story → Setting Theme Plot Resolution	Attempt → Event | Episode
Setting → State	Outcome→ Event | State
Theme → Goal | Event Goal	Resolution → Event | State
Plot → Episode	Subgoal → DesiredState
Episode → Subgoal Attempt Outcome	Goal → DesiredState

Table 3. Thorndyke grammar

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. Metrics are proposed to evaluate the validity of story candidates.

3 A Toolkit of Story Structure Abstractions as Constructors and Evaluators

To achieve a meaningful comparative evaluation of the various plot generation procedures, we consider the following steps. First we establish a reference representation format of abstract units of narrative, and we map it to the different representations of narrative used by the construction procedures considered. Second we identify how these construction procedures may be adapted to act as validation procedures. Finally, we carry out a number of experiments that combine the resulting resources.

3.1 Alligning Representations

In order to compare different story generation procedures with one another, it is important that they generate outputs in a comparable representation format. To avoid the problems associated with evaluating natural language renderings of narrative [3], in this paper we opt for direct comparison of the various procedures at the level of the sequence of abstract units of representation of narrative. This requires the adoption of a common set of abstract units of representation of narrative which might be alligned with the various elements of representation employed for the different generative procedures.

We adopt as common set of abstract units of representation of narrative the set of plot elements described in [5], which presents the following advantages. First, it allows for a relatively straightforward correspondence between elements in the set and their counterparts in the original sources. This is because it has been constructed by combination of a number of prior existing sources, with one of them being Propp's set of character functions. Second, given that Rumelhart's and Thorndyke's grammars for stories are formulated at a higher level of abstraction, establishing a correspondence between the terminal symbols of those grammars and the set of plot elements may be resolved by classifying the plot elements – which are more specific – as instances of the terminal symbols of the grammars – which are more generic.

The allignment between the set of plot elements and the various alternative representations is described in Tables 4 and 5.

It is important to note that the set of plot elements is more detailed and more fine-grained than the set of Propp's character functions. The correspondence between them has been established by considering that: (1) certain character functions represent more than one plot element – such as for instance *Abduction* and *Imprisoned* being types of `villainy` – and (2) certain character functions were phrased to encompass a range of options and the finer granularity allows for distinctions that were not available originally – such as Propp's use of `hero marries` to cover both *Reward* and *Wedding*.

With respect to the terminal symbols in Rumelhart's and Thorndyke's grammars, it must be noted that certain plot elements are slightly ambiguous with respect to their classification. Examples of this are *Cross-Dressing*, which in terms of Rumelhart's grammar can be considered either an *Action* in itself or as a *Change-of-state* that results from the action, or *AnEnemyLoved* which in terms of Thorndyke's grammar can be considered an *Event* in itself – if one focuses on the moment that it happens –, a *State* – if one focuses on the animic state of the protagonist – or as a *DesiredState* – if one focuses on what the protagonist hopes for. This suggests that the particular categories being used might require careful refinement. We consider this task outside the scope of the present paper. Although we might address it as further work, we opt at this stage for accepting that certain plot elements might be classified under more than one of the available categories.

The existence of dependencies between character functions had been identified as a fundamental ingredient in the perception of the validity of a story [4]. The same happens for plot elements: *Imprisoned* calls for *Rescue*, *Pursuit* calls for *RescueFromPursuit*. Such pairs are identified into a set of dependencies that can be checked over a given sequence of plot elements. Whereas Proppian character functions were easy to pair off (`departure-return`, `struggle-victory`), the set of plot elements is more complex in two ways: some plot elements now have two possible outcomes (*Struggle* calls for *Victory* or *Defeat*), and certain types of

Plot elements	Propp character functions	Rumelhart terminals	Thorndyke terminals
InitialSituation	*	State	State
Summary	*	State	State
Aspiration	lack	Desire, Plan	DesiredState
CallToAction	hero dispatched	Action	Event
Cross-Dressing	unrecognised arrival	Action,Change-of-state	Event
Departure	departure, transfer	Action	Event
Deliverance	delivery	Action	Event
DisconnectedFromReality	*	State	State
Discovery	*	Action	Event
Disguise	unrecognised arrival	Change-of-state	Event
Epiphany	*	Change-of-state	Event
Escape	trigger resolved	Action	Event
Guidance	*	Action	Event
HighStatusRevealed	*	Change-of-state	Event
Maturation	*	Change-of-state	Event
Metamorphosis	*	Change-of-state	Event
Pursuit	hero pursued	Action	Event
Reconnaissance	reconnaissance	Action	Event
Rescue	trigger resolved	Action	Event
RescueFromPursuit	rescue from pursuit	Action	Event
Return	return, transfer	Action	Event
SomeoneLeaves	absentation	Action	Event
Transfiguration	transfiguration	Change-of-state	Event
Transformation	transfiguration	Change-of-state	Event
UnrecognizedArrival	unrecognised arrival	Action	Event
Character'sReaction	hero reaction	Action	Event
DecisionToTakeAction	begin counteraction	Action, Plan	Event
DeceptionToFitIn	*	Action, Plan	Event
ErroneousJudgement	hero reaction	Action	Event
Ill-fatedImprudence	*	Action	Event
MoralDilemmaTriumph	hero reaction	Action, Change-of-state	Event
MoralDilemmaFailure	hero reaction	Action, Change-of-state	Event
MistakenJealousy	*	Action	Event
SacrificeForAnIdeal	*	Action	Event
SacrificeForFamily	*	Action	Event
SacrificeForPassion	*	Action	Event
SacrificeOfLovedOnes	*	Action	Event
SucumbingToTemptation	hero reaction	Action	Event
TemptationResisted	hero reaction	Action	Event
Warning/ForbiddingDisregarded	interdiction violated	Action	Event
CharacterFlaw	*	State	State
BoyMeetsGirl	*	Action	Event
BoyLoosesGirl	*	Action	Event
Wedding	hero marries	Action	Event
ClassDifferences	lack	State	State
ForbiddenLove	lack	State, Action	State, Event
Inconstancy	*	Action	Event
InvoluntaryCrimesOLove	*	Action	Event
Adultery	villainy	Action	Event
AnEnemyLoved	*	Emotion	Event, State, DesiredState
CrimesOfLove	*	Action	Event
LoveShift	*	Emotion	Event
LoveTriangle	*	Action	Event
MurderousAdultery	villainy	Action	Event
One-sidedLove	*	Emotion	Event, DesiredState
ObstaclesToLove	lack	State, Action	State, Event
ParentConvinced	trigger resolved	Action, Change-of-state	Event
RecoveryOfALostOne	trigger resolved	Action, Change-of-state	Event

Table 4. Allignment between abstract units of representation of narrative (1)

Plot elements	Propp character functions	Rumelhart terminals	Thorndyke terminals
CoupleWantsToMarry	lack	Desire, Plan	Event, DesiredState
Abduction	villainy	Action	Event
Branding	branding	Action	Event
Deception	*	Action, Plan	Event
DifficultTask	difficult task	Action	Event
Disaster	*	Action	Event
ShameOfLovedOne	*	Action, Change-of-state	Event
Exposure	false hero exposed	Action, Change-of-state	Event
Forbidding/Warning	interdiction announced	Action	Event
MistakenMurder	villainy	Action	Event
Lack	lack	State, Change-of-state, Action	Event
LossOfLovedOnes	lack	Action, Change-of-state	Event
Madness	*	State, Change-of-state, Action	Event
Misfortune	lack	State, Change-of-state, Action	Event
Persuasion	*	Action	Event
Poverty	lack	State, Change-of-state	Event
Punishment	villain punished	Action	Event
Recognition	false hero exposed	Action, Change-of-state	Event
Remorse	*	Emotion	Event
Tested	test by donor	Action	Event
TheEnigma	*	Action	Event
Defeat	*	Action	Event
Villainy	villainy	Action	Event
Imprisoned	villainy	Action	Event
LessonLearned	*	Action, Change-of-state	Event
Ambition	*	Desire, Plan	Event, DesiredState
IAmWhatIAam	*	State, Emotion	Event
Complicity	complicity	Action	Event
ConflictWithAGod	*	Action	Event
Cross-RankRivalry	*	Action, State, Emotion	Event
DaringEnterprise	*	Action	Event
HatredBetweenFriends	*	Emotion, State	State
Jealousy	*	Emotion, State	State
MisunderstandingArises	difficult task	Action, Change-of-state	Event
Revenge	villain punished	Action	Event
Revolt	*	Action	Event
Rivalry	*	Action, State, Emotion	Event
Struggle	struggle	Action	Event
JudgementDeferredToAuthority	*	Action	Event
Trickery	trickery	Action	Event
Underdog	*	Action, State	Event
UnfoundedClaims	unfounded claims	Action	Event
UnrelentingGuardian	lack	Action, State	Event
Assistance	*	Action	Event
BondStrengthened	*	Change-of-state, Emotion	Event
UsefulInformation	*	Change-of-state	Event
LackFulfilled	trigger resolved	Action, Change-of-state	Event
AspirationAchieved	trigger resolved	Action, Change-of-state	Event
ProvisionOfMagicalAgent	acquisition magical agent	Action	Event
RepentanceRewarded	*	Action	Event
Reward	hero marries	Action	Event
Riches	hero marries	Action, State, Change-of-state	Event
Victory	victory	Action	Event
MisunderstandingCleared	task resolved	Action, Change-of-state	Event
Reconciliation	hero marries	Action, Change-of-state, Emotion	Event
Repentance	*	Emotion, Change-of-state	Event
Solution	task resolved	Action	Event

Table 5. Allignment between abstract units of representation of narrative (2)

action are represented at different levels of abstraction (*Villainy* is included but also *Abduction*, *Imprisoned*,...). This influences the values of the metrics applied later in the paper.

3.2 Construction Procedures

We consider the following construction procedures:

SchemaBaseline random choice of one out of the set of schemas of narrative identified in [6] transcribed in terms of plot elements.

ProppBaseline an adapted version of Propp's suggested method of selecting elements at random from the canonical sequence of character functions [3], re-formulated in terms of plot elements in the reference vocabulary (a random choice is made when one character function can correspond to more than one plot element).

ProppDependency an adapted version of the refinement proposed in [4] that extends **ProppBaseline** with restrictions that maximise satisfaction of dependencies between plot elements across the sequence and a preference for closing on plot elements used at the end of tales in the corpus.

PropperGrammar grammar-based generation using an instance of the grammar used by the Propper system [3] with a lexicon that associates plot elements to Proppian character functions (as per Tables 4 and 5).

LakoffGrammar grammar-based generation using an instance of Lakoff's grammar for Proppian tales [7] with a lexicon that associates plot elements to Proppian character functions (as per Tables 4 and 5).

RumelhartGrammar grammar-based generation using an instance of Rumelhart's grammar [10] with a lexicon that associates plot elements to grammar terminal symbols (as per Tables 4 and 5).

ThorndykeGrammar grammar-based generation using an instance of Thorndyke's grammar [11] with a lexicon that associates plot elements to grammar terminal symbols (as per Tables 4 and 5).

3.3 Evaluation Procedures

The different formalisms for story generation described in section 3.2 can be adapted to provide a diagnostic procedure that, given a sequence of plot elements, can provide a numerical score that represents some type of conformance of that sequence to the view of narrative exemplified by the approach.

Some of the construction procedures considered provide simple solutions to achieve this. The following metrics are considered:

SS similarity between the candidate sequence and the most similar of the schemas in the set of schemas of narrative identified in [6].

PS conformance to Propp's canonical sequence of character functions – as presented in [3] – applied to the adapted version of the canonical sequence – used in **ProppBaseline** above – and corrected to deal with the differences in correspondence

DS ratio of satisfied dependencies over total number of dependencies present – adapted from the metric proposed in [4], relying on the set of dependencies identified for plot elements.

E(*Cor*) considers whether the final plot element in a candidate sequence is valid as an ending, defined in terms of a corpus *Cor* – which in the present case is Propp's original set of Russian tales (*rt*).

A grammar provides a strict judgment on the validity of a given sequence, classifying it as either valid or invalid. We require a metric that indicates the degree of partial conformance of a candidate sequence to a given grammar. To this end, for any parse of a candidate plot element sequence that does not result in a valid parse, we build a *shadow tree* that uses empty place holder nodes for the top part of the grammar, until the nodes that have been identified from the input sequence can be linked to it. In any given shadow tree, there are a number of empty nodes – which are simply place holders for non-terminal symbols of the grammar for which no support has been found in the input sequence – and non-empty nodes – which correspond to assignments of non-terminal symbols of the grammar to parses of subsequences of elements found in the candidate sequence. As indicative first approximation we consider the following metric with respect to a grammar ***X***:

GR(*X*) ratio of non-empty nodes over the total number of nodes in the shadow tree.

3.4 Combining Construction and Evaluation

The set of construction procedures is run 100 times to produce 100 different candidates sequences. The set of metrics is applied to all the candidate sequences. The average results for the various construction procedures over the basic metrics are presented in Table 6, together with some basic data on sequence length. The values of all metrics are normalised over 100 for ease of comparison.

	minL	**maxL**	**Len**	**PS**	**DS**	**E(rt)**	**SS**	**GR(R)**	**GR(T)**
SchemaBaseline	4	12	8	39	40	71	100	53	39
ProppBaseline	8	18	12	74	17	64	40	63	39
ProppDependency	7	22	14	80	49	86	46	66	39
PropperGrammar	7	14	11	68	40	100	55	53	40
LakoffGrammar	6	13	8	40	43	100	44	29	24
RumelhartGrammar	4	17	8	2	7	16	26	89	40
ThorndykeGrammar	6	19	7	0	4	5	22	50	91

Table 6. Results for basic metrics obtained by different construction procedures

4 Discussion

The first columns of Table 6 refer to the length of the generated sequences of plot elements. In each case, minimum length (minL), maximum length (maxL) and average length (Len) are given. Differences in length across the sequences arise from different factors, depending on the nature of the technique employed.

The **SchemaBaseline** instantiates one of the available schemas, which are of fixed length. **ProppBaseline** and **ProppDependency** are limited by the size of Propp's canonical sequence, and variations arise from the choices made – random (**ProppBaseline**) and driven by satisfaction of dependencies across the elements chosen (**ProppDependency**). The remaining solutions apply different grammars to generate sequences. Variations in length arise from different choices over the available rules of the grammar. Both **PropperGrammar** and **LakoffGrammar** use grammars intended to capture Propp's account. **RumelhartGrammar** and **ThorndykeGrammar** use grammars intended for more generic concepts of story.

With respect to the proposed metrics, the behaviour of the different construction procedures differs widely, as expected.

For **PS**, which captures conformance to Propp's canonical sequence, the best score is **ProppDependency** (80) with **ProppBaseline** a close second (74). The metric fails to give top scores of 100 because it sometimes penalises for non-appearance of elements in the sequence that are optional. Enforcement of dependencies makes more of these optional elements appear, whenever their antecedents have already been included. The relative performance of **PropperGrammar** and **LakoffGrammar** can be interpreted as an indication that their grammars are not entirely capturing the essence of the canonical sequence, and that the grammar used by the Propper system does this slightly better than Lakoff's grammar. In this sense, **SchemaBaseline** performs reasonably well (39), indicating that there is a close relation between the canonical sequence and the plot schemas used as reference. **RumelhartGrammar** and **ThorndykeGrammar** are clearly not built to consider this aspect.

For **DS** – degree of satisfaction of dependencies between plot elements – the top performer is again **ProppDependency** (49), that incorporates this aspect in its decision processes. The surprisingly low value in this case arises from the frequent existence of multiple dependents for a given plot element, whereas in a plot each appearance of an antecedent is normally resolved by a single consequent. It is interesting to note that the procedures that rely on representation of structure (**SchemaBaseline**, **PropperGrammar** and **LakoffGrammar**) also perform reasonably well (values around 40), presumably because the structure they use captures dependencies across the elements to a certain extent. Other procedures perform poorly. The length of the plots is not an issue because the metric is normalised over the number of potential dependencies appearing in the plot.

With respect to endings as found in Russian folk tales, **E(rt)**, it is clear that **PropperGrammar** (100) and **LakoffGrammar** (100) by construction capture perfectly the typical ending of a Russian folk tale from the corpus used by Propp.

ProppDependency (86) seems to have sometimes been forced to add tailing consequents that spoil its performance. **SchemaBaseline** does reasonably well (71) and **ProppBaseline** (64) succeeds often simply by picking elements from the end of the canonical sequence. Aditional metrics for endings need to be considered.

On similarity to existing schemas, **SS**, **SchemaBaseline** (100) shines, and all the procedures based on the Proppian account fare quite well. Again, pointing towards similarities between the canonical sequence and the schemas.

On compliance to a given grammar, not surprisingly, **RumelhartGrammar** and **ThorndykeGrammar** come out as top performers on their respective grammar, with others far behind.

The description of the results in terms of averages is useful to identify differences between different procedures. However, it is also clouding the differences that may arise between plots generated by the same procedure. It is at this level that valuable insights for further work may arise.

		SchemaBaseline 18		**SchemaBaseline 03**				
PS	52	Villainy	28	Imprisoned				
DS	86	Pursuit	40	Pursuit				
E(rt)	100	RescueFromPursuit	0	RescueFromPursuit				
SS	100	Struggle	100	Struggle				
GR(R)	53	Victory	53	Victory				
GR(T)	37	Revenge	37	Maturation				
Av	71.3	Riches	43.0	RepentanceRewarded				
		ProppBaseline 34		**ProppBaseline 03**		**ProppDependency 23**		**ProppDependency 86**
PS	85	TemptationResisted	53	Adultery	86	Character'sReaction	71	Tested
DS	47	Adultery	0	Tested	67	CoupleWantsToMarry	20	ProvisionOfMagicalAgent
E(rt)	100	ProvisionOfMagicalAgent	0	Disguise	100	Departure	0	ForbiddenLove
SS	46	Tested	31	Branding	49	UnfoundedClaims	39	Branding
GR(R)	67	MoralDilemmaFailure	53	Solution	75	Struggle	53	Return
GR(T)	37	UnfoundedClaims	37	Return	58	Victory	37	Recognition
Av	63.7	Branding	29.0	Pursuit	72.27	Return	36.7	Transfiguration
		Victory		Branding		Pursuit		
		Rescue				RescueFromPursuit		
		Pursuit				Exposure		
		RescueFromPursuit				Branding		
		UnrecognizedArrival				Punishment		
		Exposure				Reward		
		Transformation						
		Punishment						
		PropperGrammar 94		**PropperGrammar 05**		**LakoffGrammar 80**		**LakoffGrammar 74**
PS	85	ForbiddenLove	57	MistakenMurder	60	Poverty	16	Interdiction
DS	58	CallToAction	10	CallToAction	84	DecisionToTakeAction	0	DeceptionByVillain
E(rt)	100	DecisionToTakeAction	100	DecisionToTakeAction	100	Struggle	100	SubmissionOfHero
SS	40	Departure	60	Departure	46	Victory	35	Villainy
GR(R)	66	Tested	41	UnrecognizedArrival	85	Reconciliation	0	DecisionToTakeAction
GR(T)	50	Character'sReaction	37	UnfoundedClaims	37	Reward	0	DifficultTask
Av	66.5	ProvisionOfMagicalAgent	50.8	DifficultTask	68.7		25.2	MisunderstandingCleared
		Struggle		MisunderstandingCleared				Escape
		Branding		hero_recognised				Reward
		Victory		Recognition				
		Rescue		Transformation				
		Return		Revenge				
		Pursuit		Wedding				
		RescueFromPursuit						
		RumelhartGrammar 52		**RumelhartGrammar 31**		**ThorndykeGrammar 58**		**ThorndykeGrammar 01**
PS	24	Madness	0	CharacterFlaw	4	InitialSituation	0	Jealousy
DS	34	CharacterFlaw	9	DisconnectedFromReality	0	Trickery	0	Ambition
E(rt)	100	RecoveryOfALostOne	0	LackFulfilled	100	One-sidedLove	0	AnEnemyLoved
SS	37	Metamorphosis	17	LessonLearned	39	One-sidedLove	16	Aspiration
GR(R)	100	ErroneousJudgement	18	Madness	58	DifficultTask	36	CoupleWantsToMarry
GR(T)	37	Reconciliation	37	Epiphany	100	ClassDifferences	56	AnEnemyLoved
Av	55.3	LoveShift	13.5	Guidance	50.2	Riches	18.0	Warning/ForbiddingDisregarded
		Revenge		Ambition				Adultery
				Aspiration				Return
				SacrificeForPassion				Summary
				Persuasion				ClassDifferences
								Jealousy

Table 7. Examples of specific plots with values for metrics

Examples of specific plots, together with their values for the given metrics, are shown in Table 7. For each construction procedure, the best and worst performers in terms of the average of all the metrics are shown. This can serve to show a wide range of possible values without cherry-picking interesting outputs. For each example, the first column indicates the values obtained by the story on the metrics, and the second column indicates the sequence of plot elements produced.

The results in in Table 7 show a number of interesting insights. The basic structure of Propp's account is shared by many of these solutions, resulting in frequent appearance of similar sub-sequences across samples that score highly. The Rumelhart and Thorndyke grammars result in sequences that score very low on all the other metrics, but which have more surprising plot elements, combined in ways that differ from the basic sequence. Part of the problem here, is that only the syntactic part of these grammars has been considered. If the semantic interpretation rules provided by Rumelhart were considered to inform the process of selecting plot elements to instantiate the grammar, better results may be obtained. This will be considered as further work.

The differences in performance across the different approaches and the fact that now example performs well under all of the metrics indicate that each approach is focusing on a particular valuable aspect of stories. This suggests that an ideal method for plot generation should strive to combine the different aspects in a single constructive procedure. Alternatively, a simpler way of improving results might be achieved by using the metrics for some approaches to select best performers out of the set of results obtained by a different approach. This is a particularly interesting insight that will be pursued in further work.

It would be interesting to extend this evaluation to plot generation solutions beyond those developed by the author. Because the methodology invovles grounding the representation used on the reference vocabulary, this would require not just access to the source code of such solutions but also a relatively detailed understanding of the particular solution, to avoid betrayals of its spirit.

5 Conclusions

The analysis of the comparative evaluation shows that each type of procedure for the generation of stories focuses on features that may be necessary in a story. But such features are generally not sufficient, in the sense that other attempts to formulate the structure of stories with different tools may be capturing additional features that are also relevant. The comparative evaluation has served to identify a number of shortcomings in the various approaches when considered individually. Refinements of the approaches and consideration of additional approaches are possible lines of future work. However, the most promising avenue of work for short-term improvement of results would be joint use of a given generative procedure and validation procedures based on different aspects of stories.

Acknowledgements

This paper has been partially supported by the IDiLyCo project (TIN2015-66655-R) funded by the Spanish Ministry of Economy, Industry and Competitiveness.

References

1. J. B. Black and R. Wilensky. An evaluation of story grammars. *Cognitive Science*, 3(3):213–230, 1979.
2. S. Bringsjord and D.A. Ferrucci. *Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine*. Lawrence Erlbaum Associates, 1999.
3. P. Gervás. Propp's Morphology of the Folk Tale as a Grammar for Generation. In *Workshop on Computational Models of Narrative*, Universität Hamburg Hamburg, Germany, 2013. Schloss Dagstuhl.
4. P. Gervás. Reviewing Propp's story generation procedure in the light of computational creativity. In *AISB Symposium on Computational Creativity, AISB-2014*, Goldsmiths, London, UK, 2014.
5. P. Gervás, R. Hervás, C. León, and C.V. Gale. Annotating musical theatre plots on narrative structure and emotional content. In *Seventh International Workshop on Computational Models of Narrative*, Kravov, Poland, 2016. OpenAccess Series in Informatics.
6. P. Gervás, C. León, and G. Méndez. Schemas for narrative generation mined from existing descriptions of plot. In *Computational Models of Narrative*. Schloss Dagstuhl, 05/2015 2015.
7. G.P. Lakoff. Structural complexity in fairy tales. *The Study of Man*, 1:128–150, 1972.
8. R. Raymond Lang. *A Formal Model for Simple Narratives*. PhD thesis, Tulane University, 1997.
9. V. Propp. *Morphology of the Folk Tale*. Akademija, Leningrad, 1928.
10. D. E. Rumelhart. Notes on a schema for stories. *Representation and Understanding: Studies in Cognitive Science*, pages 211–236, 1975.
11. P. W. Thorndyke. Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, 9:77–110, 1977.