Functional Unit Analysis: Framing and Aesthetics for Computational Storytelling

Sven WILKE, Leonid BEROV [1]

Institute for Cognitive Science, University of Osnabrück, Osnabrück, Germany

Abstract. The plots of narratives can be analyzed by decomposition into functional units (FUs): complex structural elements that carry semantic meaning. A manual approach to the identification of FU using a graph-representation of plot has been proposed for human-made narratives, and on this basis used for plot summarization. A computational utilization, however, has remained inviable due to the complexity of the involved interpretative tasks. The present paper explores the use of the FU approach for the analysis of computationally generated narratives, because in this case interpretation can be evaded. It demonstrates that an agent-based approach to computational storytelling can be used to generate and aggregate all necessary plot phenomena in order to identify FUs. This is validated in-vivo by demonstrating that the extracted FU structure can be used to summarize, and aesthetically evaluate a plot; as has been predicted by the underpinning narrative theory. We argue that such an analysis is beneficial because it provides the system with an abstract, functional understanding of the generated artifacts, a step towards more-general intelligence. For instance, in the computational creativity domain it allows a system that has been formerly only capable of expression-type generative acts to also perform aesthetic- and framing-type generative acts.

Keywords. computational creativity, aesthetics, framing, computational storytelling, summarization

1. Introduction

Apart from generating novel artistic artefacts, computational systems need to perform other tasks in order to be deemed creative [7]: Such systems strongly benefit from being able to evaluate their output and from presenting it in a special light. This means that besides purely generative tasks they also need to perform analytical tasks. The present paper suggests that in the context of computational plot generation this can be achieved through the use of functional-unit-based plot analysis.

1.1. Functional Unit Based Plot Analysis

The functional unit (FU) model was proposed as a tool for the computational summarization of existing narratives by Lehnert [13]. It operates on a graph representation of plot and works by identifying strategically significant portions of the plot called complex FU, which are points of high relevance for summaries. Plot graphs, in this model,

[1] Corresponding Author: `lberov@uos.de`

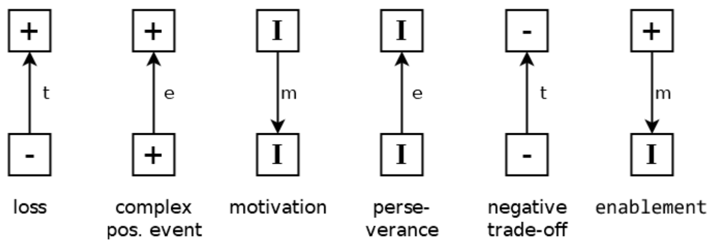

Figure 1. A sample of primitive units adopted from [13]. **I** denotes an intention, **+** and **-** are vertices of positive and negative affect.

can contain three different types of vertices, representing affect states of characters' perceptions of the events of the plot, and connected by different types of edges. The affect states that can be contained in a FU are: positive (denoted: $+$), negative ($-$) and neutral affect (*I*). Positive states describe any event which is appraised with positive emotions by a character. Inversely, negative states describe events which are negatively appraised. States of neutral affect indicate "mental states"; in this formalization always intentions. Edges describe how these states are related to each other, and can be of the following types: motivation, actualization, termination, equivalence or inter-character edges. Based on this formalism, [13] defines "15 legal pairwise configurations" (primitive FUs, e.g. 'motivation' or 'loss', see Fig. 1) that act as an alphabet, as well as a potentially open set of complex units (e.g. 'denied request' or 'retaliation', see Fig. 2) which are assembled from them and were defined based on a narratological analysis. An existing story is analysed by transforming the story-text into the introduced graph-representation, and then detecting all FUs contained in it. When this is done, a *connectivity graph* is built by using the different instances of FUs as vertices, and connecting them with edges wherever two unit instances share one or more vertices in the plot graph.

Lehnert proposed a procedure for generating summaries based on connectivity graphs [13]. For that, the units contained in the graph get translated into natural language by using template-like *generational frames* which are supplied to the program for each unit type. Into the frames, information about the specific instance of a unit is fed, allowing the frames to generate text including the characters involved in the unit, and the concrete context of the instance.

An attempt at implementing this procedure was recently made [12] with modest results, which demonstrate the complexity involved in translating narrative text into the proposed graph representation. The main problem being that the required natural language processing includes complex interpretative tasks like event-based discretization, intention and emotion detection as well as the identification of causality relations—not commonly addressed in research.

1.2. Use in Computational Storytelling

In a computational storytelling system, plots are generated by the system itself so that all required phenomena can potentially be inferred from computational representations instead of being laboriously gleaned from a text. Ideally, the system's internal representation of events already uses a graph-based format, which can lend itself for the identification of FUs. This is the case with the simulation-based storytelling system presented

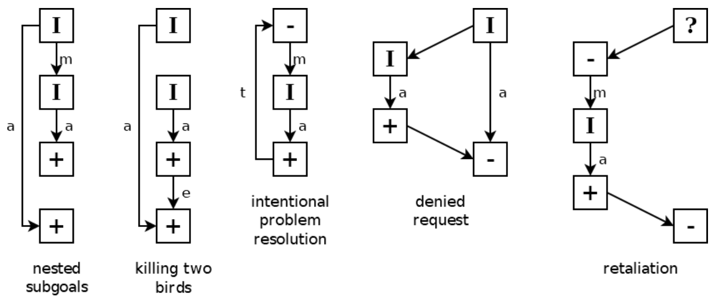

Figure 2. Examples of complex FUs adopted from [13]. '?' represents wildcard vertices.

in [1], and indeed a theoretical reformulation of the concepts required for plot unit analysis in terms of the system's architecture has been proposed in prior work [2].

Aside from their original purpose, FUs can also provide valuable information about the aesthetics of a generated plot. As proposed by Ryan [16], the strategic significance captured by FUs is important for the so-called *tellability* measure of plot quality. Tellability, here, is understood as the suitability of a set of events to be rendered as a story, i.e. a pre-textual measure of a plot's aesthetics. In its complete version it covers different aspects like motivic, contextual and formal properties. Ryan's focus is mainly on the last, which again consists of multiple factors such as semantic opposition, semantic symmetry but also *functional polyvalence*. Functional polyvalence is given, when the same event fulfills several functions in the overall plot. It can be identified through a FU analysis, by simply searching for events that are part of several complex units. Therefore, identifying the amount of overlap of FUs in a story allows an insight into the quality of its plot.

The goal of this paper is to present our implementation of FU based plot analysis in the above mentioned storytelling system. We argue for the usefulness of this analysis-method by demonstrating with a case study that it can be used to perform a cautious first aesthetic evaluation of plots and to automatically generate functional summaries, which can be used as framing. For the proof-of-concept in this paper most parts of the natural language generation is hard-coded, however, nothing precludes an extension with more linguistic knowledge at a later stage of the project.

2. Related Work

To outline the motivation for this project related work has to be introduced on the topic of computational creativity, while the specific technical contribution has to be contextualized in the field of textual summarization.

2.1. Computational Creativity

Computational creativity research is concerned with "computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative" [8]. The different tasks ("generative acts") creative systems can perform have been analysed to fall into four categories: creating exemplars, creating concepts, executing an aesthetic evaluation and creating a framing for the generated artefact ("the FACE model") [7].

Framing is contextual information that is created by an artist to put their work in a certain context, and has been known to increase an artifact's aesthetic value as well as the creativity that is ascribed to the generating process. In 'the wild' it has been known to vary wildly in form [4]. For computational systems, [4] postulated that framing can be an outline of the system's motivation, describe its intention when creating the artwork, or provide insights into how the artwork was created. We argue that, additionally, framing may also be concerned with presenting the artwork itself in a special light, that is, *framing-as-something* (the most notable example probably being Duchamp's piece 'Fountain', a ready-made urinal framed as a fountain). Based on this extension we suggest that in the case of computational storytelling, one potential piece of framing information could be a short interpretative summary of the plot, outlining abstract details like formal structure or theme-like higher meanings. An early example for work towards this direction can be found in the system Minstrel [19] that provided a moral for each story. Our work follows in this vain but differs in that it focuses its summary on laying out the functional structure of the plot (i.e. form) instead of its meaning (i.e. theme).
It has been argued that *aesthetic appreciation* is one of the three principles that guide observers in their ascription or non-ascription of creativity [6]. Indeed, an aesthetic evaluation of generated artifacts is important for a system in order to guide its generation and perform a selection of the valuable pieces from the bulk of its generation. This is a hard problem, since in most artistic domains no universally accepted aesthetic theories prevail, let alone a quantifiable evaluation function. Some systems have instead opted to quantitatively approximate reader-oriented notions like suspense [5] or tension [15]. Our work instead focuses only on the structure of the generated artefact and attempts to capture a notion of aesthetics brought forward in the context of narrative theory in order to capture the elegance and cohesion of a plot.

2.2. Text Summarization

Two main types of tasks exist for text summarization: *extractive summarization* aims at extracting the main information-bearing sentences in the source text, while *abstractive summarization* generates text not contained in the input based on some sort of reasoning about the content (see e.g. [11]). Most work performed in this context is concerned with the analysis of existing, argumentative text. Presently, the state of the art is achieved through the use of deep sequence-to-sequence neural networks, which are trained on large corpora of text in a supervised way [17]. Our approach differs radically in that it operates not on the surface realization of text, but rather a graph-based deep structure with clear functional semantics. Furthermore, it does not extrapolate summarization rules based on a corpus of existing summaries, but instead by detecting the presence of a fixed set of FUs and the way they are connected. Because this implies an act of functional analysis, this approach is located in the abstractive realm of summarization. It is furthermore based on narratological analysis and thus more appropriate for the present domain than models trained on news texts.

3. Implementation

As previously mentioned, the storytelling system's architecture was already put into the context of the FU unit model. Three steps were needed from that state to generate summaries based on the FUs of the plot: (1) The instances of FUs had to be identified in the plot graph, (2) The connectivity graph had to be built from the identified instances and (3) Natural language had to be generated from the created connectivity graph.

3.1. Functional Unit Identification

The best way to enable the detection of arbitrary complex FUs is to ensure that the whole alphabet of primitive units can be detected. For this, the capturing of vertices of intentions, positively or negatively appraised events was implemented in the storytelling system as envisioned in [2]. With the addition of motivation and actualization edges in the way [2] suggested, these vertices build most of the primitive plot units. Termination edges were implemented differently, as not only events that cause a change in the belief base (i.e. positive or negative trade-off) can be the source of a termination edge, but also the change of the belief base itself (i.e. loss and resolution, see fig. 1). The authors also predicted no need for equivalence edges between intention vertices, but in order to capture the “perseverance” primitive unit along with the related complex unit “starting over”, these connections needed to be established. No equivalence edges were implemented between non-neutral affect states. Instead, they were introduced as “causality” edges, which depicts the semantics of the primitive plot units “complex positive event” and “complex negative event”, as found in [13], more closely.

The identification of FUs in a plot graph can then be performed through application of the subgraph-isomorphism algorithm to the plot graph and each FU graph. In our implementation, the VF2 algorithm was used [9]. Since for the connectivity graph not only complex, but also primitive FUs are required, the isomorphism search is conducted for both types of units.

3.2. Constructing the Connectivity Graph

For each identified unit a respective vertex is added to the connectivity graph, and edges are added between each vertex pair that represents two FU instances with overlap in at least one vertex in the plot graph. As suggested in [13], the complexity of the graph is reduced by removing all units that are entailed by another unit, which leaves all *top-level* FUs in the graph.

While this helped to reduce the number of instances immensely, resulting graphs still contained small numbers of FU clusters with no interconnection. This arises from the properties of the storytelling system, which generates actions for each character even ‘outside of the main story line’. Thus, all instances are removed from the graph, which are not in a cluster with (i.e. directly or indirectly connected to) at least one complex FU instance.

In the original proposal, the connectivity graphs were undirected. We found it helpful to make the edges directed and let them represent temporal precedence. This allows summarizing in a temporally ordered fashion. As FU instances usually cover a time span, the execution step at which the first vertex of an instance occurred is chosen to temporally compare instances to each other (see Fig. 4 in the Appendix).

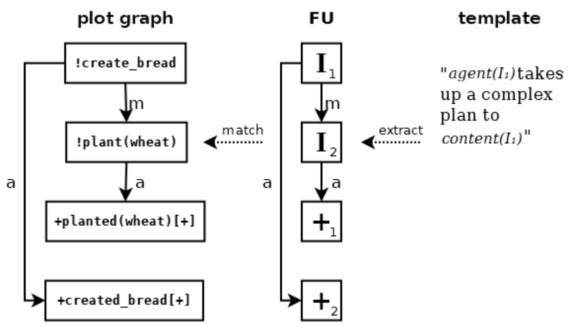

Figure 3. An example of a complex FU (nested subgoal) that matches a set of vertices in the plot. The corresponding template extracts information from the plot graph, combines it with the FU's semantics and results in a textual representation of the match.

3.3. Natural Language Summary Generation

The natural language generation algorithm was implemented prototypically, with orientation towards realizing the surface form of our case study. Better generalization to other graphs can be achieved by replacing instances of hard-coded text with linguistic knowledge.

First of all, the system selects which FU instances are to be included in the summary. For this, all instances of complex FUs are ordered with regard to the number of incident vertices (neighbors) in the connectivity graph. Then, the instance of each unique FU that has the most neighbors is chosen as the most relevant instance of that unit. Afterwards, these instances get ordered temporally, such that the resulting summary will follow (roughly) the same order as the story.

After these units have been selected, they are each translated into natural language. For this, a natural language template has been setup for each FU. The templates are able to include semantic information which was gathered from the plot graph. This includes information about which characters participated in a FU, as well as the content of one of the vertices which belongs to the unit. Which vertex' content gets extracted depends on the FU and was determined manually beforehand.

An example for how a FU maps to vertices of the plot graph is shown in Figure 3. Together with the natural language template for the unit the system can create a natural language snippet for this instance of the FU.

With this in place, the full text for each FU can be generated. The last step consists of connecting the natural language snippets of all translated units into a single sentence, by connecting them via and-conjunction. Example summaries in the context of our case study will be presented in the Case Study section.

3.4. Tellability Computation

As outlined above, functional polyvalence (an event fulfilling several functions for the plot) is taken to be one factor of tellability. This property is given, when a vertex belongs to more than one complex FU. We can therefore compute a plot's functional polyvalence fp by counting the number of vertices which belong to more than one complex FU, and dividing this number by the total number of vertices in the plot:

$$fp(G) = \frac{\sum_{v \in V} p(v)}{|V|}$$

with $G = (V, E)$ denoting the plot-graph and the function p (for polyvalence) returning 1 iff vertex v is part of more than one FU, and 0 otherwise. The normalization allows a comparison of plots of different length with regard to tellability.

Since the underlying narrative theory does not provide information on the precise quantification of tellability, this computation is subject to further empirical evaluation. However, such a measure already allows a rudimentary generate-and-test exploration of a plot space. For instance, in the employed storytelling system plot can be changed by altering the characters' personalities, and functional polyvalence can be used to compare the resulting plots. The results of such an exploration in the context of our case study will be discussed in the next section.

4. Case Study

The usefulness of the above implementation is explored via a case study on the story "The Little Red Hen". Our storytelling system is capable of recreating the plot of this fable by simulating the affect and interactions of the involved characters, and can explore the plot space spanned by this narrative universe by changing the characters' personality parameters (see [2,3]).

Three variants of the story were chosen as exemplary versions[2]. The original fable achieved the highest tellability score of the three with 0.059. In the second variant, where the hen does not punish the other animals for rejecting her requests for help but instead shares the bread, the computed tellability was 0.023. The last version contains no interaction between the animals: The hen bakes the bread alone, and eats it alone, without ever asking for help. This version scored lowest of the three, with a tellability score of 0.005. This ordering is compatible with our intuitions.

The tellability measure was also used in a generate-and-test search, where personality parameters were varied systematically and the resulting plot's tellability was measured. For computability reasons the search space had to be reduced by assuming that the hen has a distinct personality, but the other animals have the same. This revealed that tellability shows strong signs of locality, with hard boundaries. We assume that these boundaries arise from the discretization of the mood and personality traits in the agent's plan library, where behavior changes qualitatively only among the discrete values of high, medium, low, positive and negative.

Generating summaries from the connectivity graphs of the fables required further pre-processing. As the stories contain symmetry between the characters, it was more difficult for the algorithm to create a summary from the graph, because of the amount of similar but distinct vertices. We hence implemented a fairy-tale specific symmetry-detector which detects and merges vertex-pairs of the same FU belonging to different characters. Essentially, we are assuming that these FU (and character-) instances are equivalent for the summary. This resulted in the following summaries for the three story variants:

[2] A full natural language version, along with each story variant's plot graph, can be found at `http://www.home.uni-osnabrueck.de/leberov/tlrh_versions.htm`

1. I wanted to write a story in which the hen takes up a complex plan to create bread, the pig, the cow and the dog deny the hen's request for help and the hen retaliates against the pig, the cow and the dog by punishing them.
2. I wanted to write a story in which the hen takes up a complex plan to create bread and the pig, the cow and the dog deny the hen's request for help.
3. I wanted to write a story in which the hen takes up a complex plan to create bread.

5. Conclusion

This paper has presented a prototypical implementation of FU analysis in a computational storytelling system. The system was enabled to create a graph-representation of plot that captures events representing intentions, positive and negative affects as well as their interrelations of the types: motivation, actualization and termination. Based on this representation, the detection of primitive and complex FUs (structurally significant building blocks of the plot) was enabled using a subgraph-isomorphism based search. A case study was employed for in-vivo evaluation, where the presented system was applied to the plot of a fable.

Several difficulties were uncovered. A first was that inter-character edges are required not only for units that cover speech acts, but also events that are simultaneously perceived by several characters. This functionality is not easily enabled in the underlying storytelling system, which resulted in the inability to detect significant complex units. More relevantly, the case study raised our suspicion about the robustness of the approach. The complex FUs defined by Lehnert were not in all instances capable of matching the intended semantics in the automatically generated plot graphs, and hence had to be redesigned. This is not per se problematic, since their original creation was arbitrary in nature. Yet, it demonstrates that the same semantic meaning can be matched by a multitude of possible FUs, a clear sign of poor generalization. To combat this problem we implemented several modifications, like polyemotional vertices and wild-card edges, yet generally suspect that a more flexible formalism is advisable.

In general we evaluate the case study positively. FU analysis enabled the system to create plausible abstract summaries of story plots, and an aesthetic analysis of these plots based on functional polyvalence returned results that are compatible with our intuition. We thus suggest, that FU based analysis can be gainfully employed in the computational creativity context to perform framing and aesthetic evaluation tasks. To be able to benefit from this method in a comparable way, other storytelling systems need to represent characters' affect, intentions, and the causal connections between them. However, we do not consider this a drawback since the importance of these properties for understanding plot is grounded in narrative theory (apart from Ryan also compare [14,10]).

Future work is in preparation to empirically evaluate the viability of the generated summaries for framing, as well the influence of this framing on a systems perceived creativity. Furthermore, a complete implementation of the tellability measure will be performed in order to serve as quality measure in an Engagement-Reflection type [18] creative cycle. The suitability of this measure for guiding the generative process can be then empirically evaluated by comparing the resulting stories' quality, and will hopefully shed some light on Ryan's take on this elusive phenomenon.

6. Acknowledgment.

The second author is grateful for support for this work provided by an Alexander von Humboldt Ph.D. fellowship.

References

[1] Berov, L.: Steering plot through personality and affect: An extended BDI model of fictional characters. In: German Conference on Artificial Intelligence. pp. 293–299 (2017)

[2] Berov, L.: Towards a computational measure of plot tellability. In: Proceedings of the 10th INT. pp. 169–175 (2017)

[3] Berov, L., Kühnberger, K.U.: An evaluation of perceived personality in fictional characters generated by affective simulation. In: Proceedings of the 9th ICCC. pp. 24–31 (2018)

[4] Charnley, J.W., Pease, A., Colton, S.: On the notion of framing in computational creativity. In: Proceedings of the Third ICCC. pp. 77–81 (2012)

[5] Cheong, Y.G., Young, M.: Suspenser: A story generation system for suspense. IEEE Trans. Comput. Intell. AI Games **7**(1), 39–52 (2015)

[6] Colton, S.: Creativity versus the perception of creativity in computational systems. In: AAAI Spring Symposium: Creative Intelligent Systems. pp. 14–20 (2008)

[7] Colton, S., Charnley, J.W., Pease, A.: Computational creativity theory: The FACE and IDEA descriptive models. In: Proceedings of the Second ICCC. pp. 90–95 (2011)

[8] Colton, S., Wiggins, G.A.: Computational creativity: the final frontier? In: Proceedings of the 20th ECAI. vol. 12, pp. 21–26 (2012)

[9] Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: A (sub) graph isomorphism algorithm for matching large graphs. IEEE Trans. Pat. Anal. Mach. Intell. **26**(10), 1367–1372 (2004)

[10] Dannenberg, H.: Coincidence and Counterfactuality. University of Nebraska Press (2008)

[11] Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. Artif. Intell. Rev. **47**(1), 1–66 (2017). https://doi.org/10.1007/s10462-016-9475-9

[12] Goyal, A., Riloff, E., others: A computational model for plot units. Comput. Intell. **29**(3), 466–488 (2013)

[13] Lehnert, W.G.: Plot units and narrative summarization. Cog. Sci. **5**(4), 293–331 (1981)

[14] Palmer, A.: Fictional Minds. University of Nebraska Press (2004)

[15] Pérez y Pérez, R., Sharples, M.: Mexica: A computer model of a cognitive account of creative writing. J. Exp. Theor. Artif. Intell. **13**(2), 119–139 (2001)

[16] Ryan, M.L.: Possible Worlds, Artificial Intelligence, and Narrative Theory. Indiana University Press (1991)

[17] See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. ArXiv e-prints (2017), `https://arxiv.org/abs/1704.04368`

[18] Sharples, M.: How we Write: Writing as Creative Design. Psychology Press (1999)

[19] Turner, S.R.: Minstrel: a computer model of creativity and storytelling. Ph.D. thesis, University of California at Los Angeles, Los Angeles, CA (1993)

7. Appendix

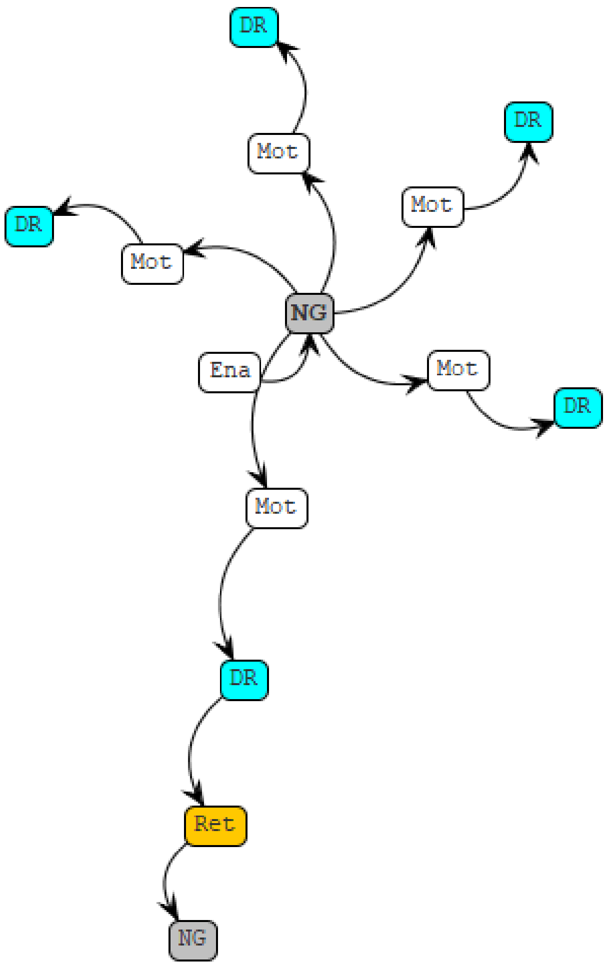

Figure 4. The connectivity graph of the original version of "The Little Red Hen". *NG* is a nested goal unit, *DR* a denied request unit, *Ret* a retaliation unit and *Mot* and *Ena* are the primitive units of motivation and enablement. The directed edges indicate temporality. The unit with the bold label has the most neighbors in the graph.