The Ideal ReaderBot: Machine Readers and Narrative Analytics

David E. Millard
University of Southampton, UK
dem@soton.ac.uk

Charlie West-Taylor
University of Southampton, UK
cwt1g14@ecs.soton.ac.uk

Yvonne Howard
University of Southampton, UK
ymh@ecs.soton.ac.uk

Heather Packer
University of Southampton, UK
hp3@ecs.soton.ac.uk

ABSTRACT

When Artificial Intelligence is considered in the context of creative activities, it is normally in the role of the creator (e.g. deep learning algorithms applied to poetry or painting). In this paper we instead propose casting the machine in the role of a reader (a ReaderBot), to help authors see how their work was likely to be experienced by an audience. This seems especially useful with interactive narratives, where multiple paths can produce many thousands of variations on that experience. We present an exploratory experiment based on the StoryPlaces locative narrative system, showing how a Simple Heuristic ReaderBot can simulate readings of a located hypertext. We then provide Narrative Analytics of those readings in the form of structural, experiential/dramatic, and locative feedback. We also present three Machine Learning ReaderBots (Linear Regression, Logistic Regression, and a Feed Forward Neural Network), trained on real reading logs, and using distance, prior visits, altitude, proximity to POIs, and text similarity as an input vector to predict next node decisions with precision substantially better than random, and comparable to the Heuristic Reader. We argue that ReaderBots can create an instant audience of thousands that could give authors valuable insights into the potential experiences of their readers.

CCS CONCEPTS

• **Human-centered computing** → **Hypertext / hypermedia**; • **Computing methodologies** → *Supervised learning by regression*;

KEYWORDS

Narrative Analysis, Machine Learning, Artificial Intelligence

ACM Reference Format:
David E. Millard, Charlie West-Taylor, Yvonne Howard, and Heather Packer. 2018. The Ideal ReaderBot: Machine Readers and Narrative Analytics. In *Proceedings of Narrative and Hypertext Workshop, held in concjunction with ACM Hypertext (NHT'18).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Authors are often advised to write with an *ideal reader* in mind, to help them focus their writing on a particular audience. The idea relates to the notion of the *implied reader* in critical theory, especially interpretations where this is a fictitious reader held in the mind of the author at the time of writing [15].

In this paper, we suggest that a useful application of artificial intelligence would be to simulate the Ideal Reader (or at least their navigational behavior) in order to give an author valuable insight into how their story might be read and experienced. This would be especially valuable in interactive narratives and hypertexts, as they have multiple branches and paths through the text. An *Ideal Readerbot* could generate many thousands of simulated readings, showing authors a probabilistic distribution of paths and experiences.

Artificial intelligence is normally applied in the role of creator (e.g. the generation of poetry [8] or music [9]), a field known as Computational Creativity [3]. But not all methods remove the author entirely, some support them in the creative process (known as *mixed initiative approaches*) [10]. However, we are proposing switching roles altogether and using AI as the *consumer* of creative outputs. There are precedents, in particular in the AI and Games community, where machine learners are taught to play a variety of games, from the challenge of General Game Playing (using games specified in formal languages such as GDL) [11], to creating artificial players for commercial games such as Angry Birds [14].

Competitive/collaborative AI has also been applied to narrative. Versu is a text-based platform for interactive drama that unfolds like an improvised play [4]. Characters in Versu can be played by a human participant or by agents. Here machines are playing the game but in order to create a rich environment for human players.

Rather than look at how an AI could replace an author, beat a human player, or collaborate with them to create a more interesting experience, our work instead looks at how an AI could support and augment human-authors by giving them instant feedback on how their creations might be read by a particular audience.

We use the StoryPlaces system as our testbed. StoryPlaces is a locative hypertext system [7] built on a sculptural hypertext engine, and with a location-aware web front-end. The StoryPlaces project has generated a number of locative stories in different locales and has worked with authors and readers to understand the poetics of locative storytelling. This gives us a starting point to create ReaderBots that mimic real human behaviors.

2 NARRATIVE ANALYTICS

Interactive stories are complex to create. Not only does an author have to deal with all the problems associated with a linear story - character and plot arcs, world-building, effective characterizations, etc. - but they also have to deal with the branching structure of the

NHT'18, July 2018, Baltimore, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

story, the agency of the reader over the events depicted, and the explosion in complexity that can result.

Authors deploy a number of structural patterns to make this manageable, including *foldbacks* (where the path forks only to rejoin later) [2], *cycles* (returning to an earlier point in the structure) [2], and *phases* (writing in specific sections that scope the story - such as scenes in Inform7 or chapters in StoryPlaces) [5].

They also have to consider all of the possible paths through their story, to make sure it is coherent, and that the story still communicates something worthwhile to the reader (although that might be different on different readings, or re-readings).

But even a simple story soon becomes unmanageable. Imagine a short story comprised of 10 nodes of 350 words each. Now consider an interactive version. If at every stage the reader had two choices leading to two different nodes, then this structure - experienced by every reader as a 3500 word story with 9 decisions - actually requires the author to create 1023 nodes, and write 358,050 words (slightly more than *Anna Karenina*). Perhaps worse than the sheer size of this text, the author also has to manage 511 reader choices, and consider 512 different reading paths, each with its own ending.

No wonder then that many interactive stories actually deploy a relatively simple structure, especially when content is expensive to produce. For example, many narrative driven video games are linear, with the interaction purely unlocking progress (for example, Naughty Dog's *The Uncharted Series*), or use a sequence of foldbacks to return to a core path, branching at the very end for a small number of alternative endings (for example, BioWare's *Mass Effect*). In classic hypertext fiction and modern interactive fiction content is typically textual, and structures can be more easily ambitious: some heavily branching but shallow, others using complex foldbacks, others following a topological model (allowing navigation and revisitation between a maze of nodes) [1].

Ironically, some of these approaches, which are designed to reduce the overall number of nodes required, actually increase the number of choices and paths. Any cyclic network has effectively an infinite number of paths, but even acyclic networks that create connections between branches (including foldbacks) magnify the number of possible reading paths. Size is therefore reduced, but complexity is not. Authors are still required to keep all of these possible paths in mind when planning and writing.

Calligraphic writing tools often include graphing tools to lay out and plan structure (for example, *Twine* or *StorySpace 3*), but sculptural tools (for example, *StoryPlaces* or *StoryNexus*) use rules and constraints to dictate choices, resulting in emergent runtime graphs with the potential to be much denser than calligraphic ones (and therefore more complexity in terms of paths and reader choices).

We present Narrative Analytics as a potential way to help address these problems. By analyzing the dynamics of the system (as dictated by graph or rules/constraints) an analytical system could support an author in spotting issues with their interactive narrative. One approach would be to undertake an analysis of the graph itself, however, this has the problem that it lends equal weight to each choice, whereas in practice human readers will favor certain choices over others. For example, in a locative hypertext, readers tend to make choices that are nearer to them in physical space [12]. Therefore the approach that we suggest here is to use a machine reader, a *ReaderBot*, to generate hundreds or thousands of simulated readings of the story, and then analyze this set of readings, rather than analyze the graph directly. This process is shown in Figure 1.

ReaderBots could potentially be implemented in a variety of ways and could even simulate different types of readers, but their implementation is always independent of the analytics itself, which looks at the set of readings regardless of how those readings were generated. This cleanly separates the analysis methods from assumptions about how reading takes place.

Such an analysis could potentially spot problems with structure, use of locations, and the dramatic aspects of the experience.

2.1 Problems with Structure

Structural problems are possible in all hypertexts, but are more difficult to spot in sculptural hypertexts as there is no explicit graph that can be visually examined. A narrative analysis could identify a number of structural issues, for example:

Dead Ends - a state from which no further progress is possible. Some of these are intended end points of the story, but others are loose ends that need to be repaired. Some systems (e.g. StoryPlaces) explicitly flag nodes that end the story, meaning that genuine dead ends can be identified more easily.

Unreachable Nodes - a node (or set of interconnected nodes) that are isolated from the network and therefore never appear in any reading.

Unbreakable Cycles - a cycle of nodes where there is no opportunity to break the cycle (although some stories use this intentionally and have no intended end point). Unlike a traditional analysis, ReaderBots could also spot breakable cycles where the break is hard to find - by comparing the proportion of readings that break the cycle to those that do not.

2.2 Problems with Experience and Drama

Although a full critical reading of each path generated by a ReaderBot is beyond the ability of the machine, an approximation could be undertaken by using content analysis methods, or more simply by allowing an author to tag certain nodes and then looking for the appearance of these tags in the reading set. Factors might include:

Length - a distribution of the word lengths across the readings. Useful for understanding the likely range of experiences, and for spotting abnormal path lengths in the set.

Language - comparing the consistency of language across the readings (e.g. using Linguistic Inquiry and Word Count [13]).

Theme - looking at the themes present in different readings, e.g. using topic analysis or thematic analysis [6].

Key Information - such as plot points or character revelations could be tagged by the author, and the readings examined to see whether they were present, or in an appropriate order (e.g. Chekhov's Gun, where a weapon seen earlier in the story creates an expectation that it will be fired later).

Dramatic Elements - such as meaningful events or scenes could also be tagged, to check how they are distributed throughout the readings, and look at the impact on pacing.

POV - in stories with multiple threads or Points of View (POV), the analyses could reveal what proportion of time different readers spent with each part of the story.

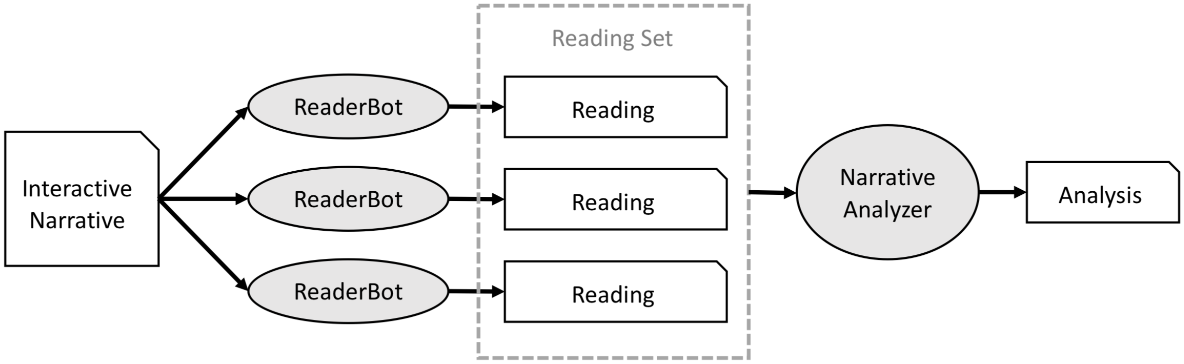

Figure 1: The Narrative Analytics Process using ReaderBots

2.3 Problems with the Use of Locations

In locative stories, the physical environment and topology is an important factor in the experience of readers. One of the outcomes of StoryPlaces was a Toolkit - a simple guide for locative authors for how to best use the environment and avoid common mistakes [12]. Some of these rules could be checked automatically, for example:

Points of Arrival/Departure - should be close together; stories may have multiple starting points and multiple ending points, but an analysis of the reading set will show the distribution of distances between start and end of all the readings.

Reader's Efforts to Move - should be minimized; by avoiding zig zags and double-backs. These can be detected automatically by looking for sub-chains of nodes in the reading set which start and end in similar locations.

Total Time to Read - needs to match the visitor expectations for that area (if visitors typically have an hour to spare then stories must fall within this), this can be calculated using assumptions about total distance traveled and walking speeds.

High Cost Locations - such as those requiring a long detour, or for the reader to climb a hill, should be used sparingly. This can be checked by looking for node traversals that are unusually long, and by comparing altitude data for nodes.

Points of Interest - should be integrated into the experience as they naturally draw attention. POIs can be identified from public datasets such as DBpedia, and cross referenced against the locations in the readings. A POI could appear in the story, but be rare in the reading set.

3 A SIMPLE HEURISTIC READERBOT

We developed a simple heuristic ReaderBot that makes decisions about where to go next based on two observations we have made of real readers: that they tend to visit nodes that are closer, and that they are less likely to revisit nodes that they have already read.

3.1 Implementation

More formally, we define that for every target node t where $t \in T$, there is a function $visit(t)$ that returns the number of visits to the node[1], and a function $distance(t)$ that returns the straight line distance (defined in meters based on information from the OpenStreetMap API, or set to 20 if the distance is 0 or if there is no location set[2]). We then define a weight for each t as follows:

$$weight(t) = distance(t) * 1.5^{visit(t)} \tag{1}$$

This creates a weight that grows linearly with distance and increases exponentially with visits. We then calculate the inverse with respect to the *min* and *max* of the range (as the further away and more visits the lower we want the chance of choosing to be):

$$W = \{weight(t) | t \in T\} \tag{2}$$

$$invweight(t) = max(W) + min(W) - weight(t) \tag{3}$$

We then normalize the weights across all potential target nodes:

$$chance(t) = invweight(t) \Big/ \sum_{i \in T} invweight(i) \tag{4}$$

$$C = \{chance(t) | t \in T\} \tag{5}$$

This ensures that $\sum_{i \in C} chance(i) = 1$. At each step the ReaderBot then uses a random number, mapped against the values in C, to choose the next node to visit. It records its path as a single reading. Our scripts can thus generate multiple readings through the story, creating a reading set for analysis. The code for the ReaderBot and Analysis engine is available on GitHub[3].

3.2 Analysis

We took the twelve StoryPlaces stories developed for the Southampton and Bournemouth areas and used our Heuristic ReaderBot to generate 1000 readings for each story. The results of the analysis are shown in Table 1. Time is calculated based on a walking speed of 1.4 m/s. Note, Fallen Branches does not have an ending, so we capped the reading at 100 visited nodes. we then looked at four factors (taken from the set proposed in Section 2):

[1] In StoryPlaces nodes and locations are independent, this means that a reader returning to a location to read a new or updated node does not count as a repeat visit

[2] Locations of nodes are typically set as lat/long values with a default radius of 20m, meaning that nodes nearer than 20m are effectively equivalent to a distance of zero

[3] ReaderBot on GitHub: https://github.com/charlie-wt/user-model

Table 1: Analysis of Southampton and Bournemouth Stories

	Nodes				Word Count			Distance (m)		Time (sec)			Start-End (m)		
	No.	Ur	Mean	SD	Mean	SD	Range	Mean	SD	Mean	SD	% <1hr	Mean	SD	% <200m
A Walk In The Park	5	0	8.2	1.7	2220	61	420	522	88	731	123	100	139	55	100
Butterflies	11	0	12.0	0.0	1943	0	0	967	0	1353	0	100	717	0	0
Connections	9	0	10.0	0.0	2813	0	0	1979	324	2770	452	100	468	246	21.5
Fallen Branches	65	0	100	0	21775	317	2013	2485	426	3479	597	99.7	95	0	100
Fire fire!	8	0	8.0	0.0	490	9	18	1415	0	1981	1	100	8	0	100
Naseem - Pharaoh's Attendant	11	0	12.0	0.0	1178	0	0	1228	0	1720	0	100	88	0	100
Notes on an Illegible City	12	0	12.0	0.0	1565	42	83	1156	169	1618	237	100	384	184	30.6
Seeker of Secrets	31	0	25.2	4.2	2538	420	2269	20112	5560	28157	7784	0.2	1970	1070	2.4
Six Stories Of Southampton	9	0	10.0	0.0	1997	0	0	3059	438	4283	614	87.8	895	384	3
The Bournemouth Triangle	55	0	45.1	1.0	7449	149	409	6398	73	8957	102	0	1346	0	0
The Destitute and the Alien	21	0	21.6	0.5	5080	127	263	4430	306	6202	428	0	737	0	0
The Pathways of Destiny	81	0	11.5	3.2	2930	781	3079	491	459	688	643	100	98	116	78.9
The Tale of Molly DeVito	9	0	10.0	0.0	2242	0	0	1226	0	1717	0	100	390	0	0
The Titanic Criminal	10	0	18.7	1.6	2251	0	0	1587	0	2222	0	100	736	0	0

Unreachable Nodes (Ur) - none of the stories contain unreachable nodes, although in testing we discovered that if the number of readings drops to below 500 some of the stories start to show unreachable nodes. In particular, *The Pathways of Destiny*, contains significant early branches, meaning that it behaves effectively as seven different stories, and as the readings for each branch drops below 100 it becomes more common for the ReaderBot to miss certain end nodes.

Total Text Length (Word Count) - There are two stories that show an extremely high range in word count (*The Pathways of Destiny*, range is 105% of mean, and *Seeker of Secrets*, range is 89% of mean), indicating that the reader experience is very variable. This is because both stories have an early point where the story can be ended - which may work narratively, but not necessarily for the experience.

Points of Arrival/Departure (Start-End) - A significant number of the stories show greater than 200m distance between start and end (which represents around 5 min walking). The Toolkit was developed after most of these stories were written, although it was used to create *Fallen Branches* which manages to have a modest Start-End distance despite having by far the longest word count for the experience.

Time to Read (Distance, Time) - All of these stories were intended to be read within one hour, but many fail to achieve this. One story, *The Seeker of Secrets*, was created by an author using Google Maps and StreetView, and the author was surprised when reading it for the first time in the locations. The analysis picks this out as a particularly challenging example, which on average takes nearly 8 hours for the ReaderBot to process. In reality, our observations of readers show it takes 3 hours, perhaps because real readers reduce re-reading to effectively zero when the distances rise above a certain point.

The analysis shows problems with the stories that we saw in our physical observations of readers, and with an appropriate visualization would almost certainly have been of use to our authors. However, it also shows some unrealistic behavior, which demonstrates the limitations of a rigid heuristic approach.

4 MACHINE LEARNING READERBOTS

We created a number of ReaderBots that used machine learning techniques to make decisions about the next step in a simulated reading. We extended the factors to include:

Walking Dist - the distance to the target as returned by the Open Source Routing Machine (OSRM)[4].

Visit Count - the number of prior visits to the target

Altitude - of the target, taken from the Shuttle Radar Topography Mission (SRTM)[5]

POI - the number of nearby (<100m) Points of Interest (POIs) to the target based on the Overpass API[6]

Mentions - the number of times a word from the target's title is mentioned in the current node's text (weighted by TF-IDF)

We built an input vector based on these five factors, as well as the ranking of the page on each of these measures relative to the other pages visible at that point. The target value is either 1 (page was chosen) or 0 (not chosen). There were 1126 total input vectors taken from 55 total readings across the 7 stories which contained decision points, and where we had logged readings.

4.1 Implementation

We created three seperate ML ReaderBots:

Linear regression using L2 regularisation, a gradient descent optimiser and 10-fold cross-validation.

Logistic regression using L2 regularisation, a gradient descent optimiser and 10-fold cross-validation.

Feedforward Neural Network using ReLU activation and the Adam optimiser, with 1 hidden layer of size 5.

We used the same parameter values (refined via testing) for each:

- Learning rate: 0.01
- Epochs: 100
- Batch size: 1
- Convergence threshold: 0.0001

[4]OSRM: http://project-osrm.org
[5]SRTM Long Term Archive: https://lta.cr.usgs.gov/SRTM
[6]Overpass: http://overpass-api.de

Table 2: One-Step Look-ahead Precision of the ReaderBots

	Rand	Heur	LinR	LogR	NN
A Walk In The Park	0.216	0.608	0.608	0.627	**0.647**
Fallen Branches	0.546	**0.909**	0.864	0.818	0.773
Notes on an Illeg ...	0.418	**0.791**	0.627	0.642	0.597
Seeker of Secrets	0.167	**0.542**	0.208	0.333	0.208
Six Stories Of Sou ...	**0.500**	**0.500**	**0.500**	**0.500**	0.000
The Pathways of D ...	0.125	0.292	0.292	0.292	**0.333**
The Titanic Crim ...	0.436	0.615	0.718	0.795	**0.897**
Overall	0.344	**0.608**	0.545	0.572	0.494

4.2 Evaluation

We evaluated each of the three Machine Learning (ML) ReaderBots based on their one-step look-ahead precision (i.e. at each point in the story did they correctly identify the next step as the most likely one). As the stories vary in complexity it is not meaningful to compare between stories, but we can see the performance relative to other ReaderBots. We also generated this precision value for our original Heuristic ReaderBot (as described in Section 3, and still using straight-line walking distance), and for a Random ReaderBot to give us a base-line performance for comparison. The precision results are shown in Table 2 with the best performance in bold.

Open stories (such as *A Walk in the Park* and *Seeker of Secrets* tend to perform badly with the random reader (as there are many options at each decision point), and show significantly increased precision in the Heuristic and ML Readers. *Pathways of Destiny* has low absolute values for all readers, perhaps because its decision points are more directly asking the reader to make decisions about the story (E.g. Will you help the alien?, or What role do you want to take on in the story?). Stories where decision points tend to have a limited number of possibilities (for example, *Fallen Branches* which has many decision points with only 2 or 3 possible options) tend to show much higher absolute scores.

In general, the ML ReaderBots perform significantly better than random, but not always as well as the Heuristic ReaderBot. The Neural Network manages to produce the highest precision with three stories, but is surpassed by Linear and Logistic Regression on the others (where Heuristic is Highest).

It may well be the case that certain ReaderBots perform better on certain types of story (e.g. Linear vs. Open), but that analysis is beyond the scope of this paper. However, these initial results do show that a trained ReaderBot (even on this relatively small dataset) can approximate results based on rules that come from human observation - and confirms that our choice of distance and visits as factors for the Heuristic ReaderBot was a reasonable one.

5 CONCLUSIONS

Machine learning is often considered in the context of the *creation* of digital art, but in this paper, we have argued that it also has a valuable use in simulating the *consumption* of that art. In particular when it is used to simulate a reader of an interactive digital narrative. This Ideal ReaderBot could potentially give authors valuable feedback about the likely behavior of human readers.

When coupled with analyses that look at best practice for digital authors (e.g. the StoryPlaces Toolkit), or author specific markup to highlight features within a story, the ReaderBots' readings could be used to support authors in creating more effective interactive stories, by giving them more confidence that every reader was being given whatever they define to be a quality reading experience.

This paper describes early work, but does demonstrate the potential of the approach, and shows that even a simple heuristic reader has real value. We would like to develop our machine readers further, train them on larger datasets, and evaluate them against real reading patterns in new stories. It would also be interesting to consider an alternative analysis approach, by exploring how a ReaderBot might recursively build a Markov Model of reading states, rather than a finite set of example readings, and how this might be used to power other types of analysis.

We have presented an alternative view of a mixed-initiative approach to computational creativity that shows how AI in the form of ReaderBots could support and augment human authors by simulating readers rather than by automating aspects of the authoring role. We believe that working alongside ReaderBots would give authors the confidence to create more intricate and sophisticated creations than they would working alone.

ACKNOWLEDGMENTS

The authors would like to thank Patrick McSweeney, who developed an early prototype of a StoryPlaces narrative ReaderBot; Anqi Yu, who contributed to the design of the analyser; and Nicholas Gibbins for his comments on an earlier version of the paper.

REFERENCES

[1] Sam Kabo Ashwell. 2015. Standard Patterns in Choice-Based Games. https://heterogenoustasks.wordpress.com/2015/01/26/standard-patterns-in-choice-based-games/.
[2] Mark Bernstein. 1998. Patterns of Hypertext. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*. ACM, New York, NY, USA, 21–29.
[3] Simon Colton, Geraint A Wiggins, et al. 2012. Computational creativity: The final frontier?. In *ECAI*, Vol. 12. 21–26.
[4] R. Evans and E. Short. 2014. Versu âĂŤ A Simulationist Storytelling System. *IEEE Transactions on Computational Intelligence and AI in Games* 6, 2 (June 2014), 113–130. https://doi.org/10.1109/TCIAIG.2013.2287297
[5] Charlie Hargood, Verity Hunt, Mark Weal, and David E. Millard. 2016. Patterns of Sculptural Hypertext in Location Based Narratives. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media*. ACM, New York, NY, USA.
[6] Charlie Hargood, David E Millard, and Mark J Weal. 2008. A thematic approach to emerging narrative structure. In *Proceedings of the hypertext 2008 workshop on Collaboration and collective intelligence*. ACM, 41–45.
[7] Charlie Hargood, Mark J. Weal, , and David E. Millard. 2018. The StoryPlaces Platform: Building a Web-Based Locative Hypertext System. In *Proceedings of the 29th ACM Conference on Hypertext and Social Media*. ACM, New York, NY, USA.
[8] D Fox Harrell. 2005. Shades of computational evocation and meaning: The GRIOT system and improvisational poetry generation. In *Proceedings, Sixth Digital Arts and Culture Conference*. 133–143.
[9] Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. 2017. A Functional Taxonomy of Music Generation Systems. *ACM Comput. Surv.* 50, 5, Article 69 (Sept. 2017), 30 pages. https://doi.org/10.1145/3108242
[10] B. Kybartas and R. Bidarra. 2017. A Survey on Story Generation Techniques for Authoring Computational Narratives. *IEEE Transactions on Computational Intelligence and AI in Games* 9, 3 (Sept 2017), 239–253.
[11] N. Love, T. Hinrichs, D. Haley, R. Schkufza, and M. Genesereth. 2008. General game playing: Game description language specification.
[12] Heather S Packer, Charlie Hargood, Yvonne Howard, Petros Papadopoulos, and David E Millard. 2017. Developing a Writer's Toolkit for Interactive Locative Storytelling. In *Int. Conf. on Interactive Digital Storytelling*. Springer, 63–74.
[13] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. Technical Report.
[14] Jochen Renz et al. 2015. AIBIRDS: The Angry Birds Artificial Intelligence Competition.. In *AAAI*. 4326–4327.
[15] Wolf Schmid. 2014. Implied Reader. In *Handbook of Narratology*, Pier J. Huhn P., Meister J.C and Schmid W. (Eds.). Walter de Gruyter GmbH, 301–309.