Towards Automatic Story Clustering for Interactive Narrative Authoring

Michal Bída, Martin Černý, and Cyril Brom

Charles University in Prague, Faculty of Mathematics and Physics,
Malostranské nám. 2/25, Prague, Czech Republic

Abstract. Interactive storytelling systems are capable of producing many variants of stories. A major challenge in designing storytelling systems is the evaluation of the resulting narrative. Ideally every variant of the resulting story should be seen and evaluated, but due to combinatorial explosion of the story space, this is unfeasible in all but the smallest domains. However, the system designer still needs to have control over the generated stories and his input cannot be replaced by a computer. In this paper we propose a general methodology for semi-automatic evaluation of narrative systems based on tension curve extraction and clustering of similar stories. Our preliminary results indicate that a straightforward approach works well in simple scenarios, but for complex story spaces further improvements are necessary.

1 Introduction

Development of interactive storytelling (IS) systems is a challenging process involving multi-disciplinary knowledge. Despite this, a number of working interactive storytelling systems was developed through the years. Some examples of these systems involving intelligent virtual agents are games such as Façade [1] or Prom Week [2], research projects such as ORIENT [3] or BierGarten [4], educational games such as FearNot! [5] or immersive IS systems such as Madame Bovary [6].

Evaluation of these systems is a demanding process often requiring extensive effort. State of the art evaluation of IS systems involves either surveys with users, e.g. [7], or technical evaluations such as comparing length and complexity of generated stories. Neither of these approaches is well suited for practical development. While well designed user surveys can capture the quality of generated stories correctly, they are costly and take a lot of time and thus it is problematic to make them part of a regular development cycle. Since combinatorial explosion of the story space is in most cases a desirable property of an IS system, it necessarily follows that for large-scale IS systems a thorough sampling of the story space by human users is costly at least and unfeasible at worst. While technical evaluations of the narratives are feasible for much larger story spaces than user studies and are significantly cheaper, they are only loosely related to the actual enjoyment of the stories by the user and do not allow the story author to understand what is happening in the generated stories and to shape them according to his artistic vision. In this paper we will consider a method allowing

H. Koenitz et al. (Eds.): ICIDS 2013, LNCS 8230, pp. 95–106, 2013.

for computer-assisted evaluation of stories generated by an IS system. In general, the human designer is kept in the loop, but the computer aggregates data from the system so that only relevant parts of the story space need to be inspected directly. We will focus on meaningful clustering of generated stories to groups with similar properties.

This paper is organized as follows: First, we will survey related work concerning drama analysis. Then we will describe our methodology of drama analysis and present some preliminary results on data gathered from two versions of our IS system [8] and [27]. We will conclude the paper with discussion and future work.

2 Related Work

In this section we present an overview of literature relevant to semi-automatic story evaluation. To our knowledge only little work has been done on story clustering. This overview is mostly a list of interesting ideas that could be reused in the context of automatic narrative evaluation and clustering.

In [9], Brewer and Lichtenstein propose a structural-affect theory of stories with three main aspects – surprise, suspense and curiosity. They claim that these aspects affect how much the reader likes the story. They outline how these aspects manifest in stories. In our approach we consider only tension (suspense) curve so far, however surprise and curiosity curves may be good candidates for additional story features.

In [10], y Perez and Sharples describe MEXICA, a system capable of producing emergent stories with user assistance. Among others, it uses tension curve of the stories to evaluate interestingness based on tension degradation and improvement in time. In our work we are currently only interested in overall tension curve shape.

Ware et al. [11] present four quantitative metrics describing narrative conflict. The measurement and conflict detection are based on a planning representation of the narrative, making them harder to employ in non-planning IS systems. Their evaluation showed the automatic evaluation matched user evaluations of conflicts. An interesting future work may be to detect conflicts in the story, evaluate these conflicts based on [11] and use the output as additional story features.

Weyhrauch [12] implements several evaluation functions for his emergent narrative system. These functions encode the accumulated domain knowledge of the author and hence are tightly bound with his storytelling system domain. However these functions provide a good example of what can be measured in IS and how it can be done.

In [13] Ontañón and Zhu propose an analogy-based story generation system, where they evaluate the quality of resulting stories by measuring their similarity to “source” stories (input human-made stories). The similarity metric is based on MAC/FAC model [14]. This work provides an alternate approach to similarity measure based on case based reasoning.

Cheong et al. [15] present a system capable of extracting the story plan from game logs and generating a visual story summarization. Although the system is not concerned with story evaluation directly, story abstractions created by this method could be a good input for an evaluation metric.

Schoenau-Fog [7] discusses how to evaluate interactive narratives with users by questionnaires and intrusive methods. Although the techniques are not automatic, some of the ideas may be transferrable to automatic evaluation systems.

In [16], Rabe and Wachsmuth propose an event and an episode similarity metric for their virtual character. Although they are not using this in the context of story evaluation, the features they are working with might be reused in this context.

Zwaan et al. [17] propose and test a model of how readers construct the representations of situations occurring in short narratives. They claim that the readers of stories update their mental models along five indices: temporality, spatiality, protagonists, causality and intentionality. These ideas could be used as a basis of automatic evaluation systems.

Porteus et al. [18] use Levenshtein string distance to measure differences between various stories generated by their storytelling system.

Aylett and Louchart [19] propose the use of double appraisal to increase the dramatic tension of an action. Their characters choose next actions based on their current emotions and based on the emotional impact of their actions on other characters in the story. To simulate emotions and to measure emotional impact, OCC based model is used. The difference between their and our approach is that we are using the output emotional impact to compute the tension curve and not to influence the decision making of our characters.

Kadlec et. al [20] uses compressibility and conditional entropy to measure similarity between sequences of actions gathered by human tracking and by daily corpora simulator.

Narrative evaluation is a complex topic involving general narrative and drama theories, human and computational story abstractions, algorithms allowing for comparisons and evaluations of these abstractions and data gathering methods. In our approach we have put together the following ideas presented above: a) the use of tension curve to represent the story, b) measurement of story similarity to group stories together and c) the use of string metrics to define difference between stories. The novel part of our approach is our tension curve extraction mechanism and the use of a standard machine learning clustering algorithm for grouping the stories together. Our methodology will be described in detail in the next section.

3 Methodology

We propose semi-automatic narrative analysis by a meaningful clustering of narratives into groups with similar stories. This would work as follows: a) the developer uses his system to generate large number of stories, b) the developer runs our clustering algorithm dividing the stories into groups and c) the developer now needs to see only several stories from each group to evaluate the system as the other stories in the group are similar, saving development time. Our clustering is based on two general features of stories: a) a story action sequence and b) a story tension (dramatic) curve. As a clustering algorithm, we use k-means [21] (details in section 3.2).

To evaluate the methodology we have conducted experiments on our IS system SimDate3D (SD) level one detailed in [8]. SD level one is a simple 3D dating game where the goal of the user is to achieve that a couple – Thomas and Barbara – gets to the cinema. The game comprises a sketchy conversation through comic-like bubbles called emoticons and the user partially controls one of the characters actions. There are three possible endings of this story: a) characters get to the cinema safely, b) characters get angry and part and c) characters interaction is too positive, so they decide to skip the cinema and head home. Recently, we have developed a sequel – SD level two [27] (Fig. 1) – an extended scenario, where Thomas is now dating two girls at the same time. All game sessions end with all three characters meeting and engaging in an argument. The outcome of this argument depends on previous user actions and there are four endings: a) Thomas staying with Barbara and breaking up with Nataly, b) Thomas staying with Nataly and breaking up with Barbara, c) both girls breaking up with Thomas and d) Thomas staying in the relationship with both girls.

Fig. 1. Screenshots from SimDate3D showing interactions between characters

3.1 Tension Curve Extraction

The dimension of tension was proposed by Freytag in his book [22] where he expanded the Aristotle's theory of dramatic events. Most narratives of mainstream books and movies follow one of the standardized tension curve shapes. Hence stories generated by IS systems should usually follow one of the curves as well. Indeed, tension curves have been explicitly or implicitly modeled by multiple IS systems (e.g. [1]). But how to extract a tension curve from generated stories in general? One possible way is to let the authors annotate each action or plot sequence with a tension

value. The tension curve of the story could then be constructed from individual tension values. The problem here is that the tension of actions often depends on the current story context, which makes such annotations very difficult.

On the other hand story tension is often related to emotions experienced by the story protagonists. Thus the tension could be extracted from character emotions, if the IS system does explicitly represent them. Note that this approach also has its limitations. For example consider a situation where a dark figure is following someone in a deserted part of the town. As the figure gets closer the tension in the story indeed rises. However, this may not be represented well by character emotions as the one who is followed may not be aware that someone is following him. One of the ways of solving this issue is to generate tension from emotion model simulating the user emotions directly based on what the user perceives in the story at the moment. A method involving double appraisal [19] might be a reasonable candidate.

In our IS system, characters are equipped with OCC-based [23] emotion model, which directly influences their decision making. We have defined tension in our IS system as follows: Every 250 ms we have made a snapshot of all characters' emotions. Then we took the sum of these emotions where every positive emotion was counted with a minus sign and every negative emotion was counted with a plus sign. The resulting number encoded the tension value at the moment. The tension curve is then simply the piecewise linear function defined by these values.

3.2 Clustering

To compute clusters for stories in our IS system we have used k-means clustering algorithm [21]. The k-means clustering algorithm is a method of analysis which aims to partition n samples (in our case stories) into k clusters (k groups of stories) such that the distance between stories within a single cluster is minimized. The distance is computed with a user specified metric. The k is fixed and given by developers. The k-means algorithm requires initialization of the starting position of clusters. Afterwards it works in two stages. First it assigns the "nearest" cluster to each sample. Then it moves all clusters by computing the average of members in the cluster and shifting the cluster towards this new average. These two stages are repeated until the clusters become stable or a maximum number of steps is reached.

The idea of using k-means is that if we define the "distance" between stories right, then every resulting cluster will contain stories that are somehow similar to each other. We have tested and compared four distance metrics. One was based on the tension curves of the stories and three were based on the action sequences of the stories.

To compute the distance between stories using tension curves, we simply computed the distance between tension values from one story to values from the other and summed absolute values of the differences as seen in formula (1), where x and y are tension curve vectors of respective stories and n is the length of the vectors. If the tension curves differed in length, the shorter was expanded to match the size of the larger by adding zeros.

$$\text{distance}(x, y) = \Sigma_{i=1..n} |x_{(i)} - y_{(i)}| \quad (1)$$

To compute distance between action sequences, we have first encoded the action sequences as *action strings*. The raw actions extracted from our system look like "Thomas greets Barbara", "Barbara greets Thomas", "Thomas asks Barbara to go to the cinema", etc. We have assigned a single letter to encode every action type (e.g. the action "says hi" is represented by "A" regardless of who said it). The action string is formed by concatenating letters representing the individual actions in the sequence. There are around 50 possible actions in our scenario (we used upper and lower case letters and numbers to encode them). For level two we have added one more letter per action representing the character performing the action (see 4.1 and 4.2 for examples).

To measure distance of action strings we have tested three standard string difference metrics – Levenshtein distance [24], Jaccard index [25] and Jaro-Winkler distance [26]. To summarize, the distance between stories in this case was defined by the string difference metrics working on *action strings* of the stories.

Motivation for using string metrics in the analysis is twofold. Firstly, the action sequence is a natural representation of the story. Secondly, string distance is a well studied subject and many techniques that could be directly applied to our case were tested and developed in a different field. And in case of Levenshtein distance being applied on our *action strings*, the string editing procedure which defines the metric can be perceived as a rudimentary form of story editing.

To run k-means clustering with the distance metrics presented in this section we needed to solve two issues: a) random initialization of clusters and b) computation of cluster average from its members with a given distance metric. Concerning a) the random initialization of clusters sometimes caused undesirable behavior where one or more of the clusters were so far from the samples that they stayed empty the whole time. To compensate for this we have initialized clusters as follows. For each cluster we have selected one story from the sample set at random that defined the initial position of the cluster. Concerning b) the problem was that for string distance metrics it may not be clear what an average of several action strings means. When computing the average from cluster members with string distance metric we have simply iterated over all members of a cluster and picked the member with the lowest sum of distances from the other members. This member then became the new cluster average.

3.3 Evaluation

Our goal in this stage was to evaluate the quality of k-means clustering based on story distance definitions above. We have examined the performance on two domains – SD level one and SD level two. Level one presented a simple domain and level two a complex domain. For level one we have run k-means clustering with three, four and five clusters (defining the resulting number of clusters). Level two was run with k set to four, five and six clusters. It is known that k-means clustering may converge to local optima. To account for this we have run each setting of k-means clustering eight times, e.g. k-means with four clusters and tension curve distance metric was run eight times on level one and the results were averaged.

As there is no generally accepted method for evaluating the quality of clustering independent of the application, we have resorted to ad hoc method suitable for our

scenario. Our goal in this stage was to confirm whether the k-means algorithm with a specific distance metric is capable of detecting some of the meaningful features of the input stories, i.e. whether it would group stories with similar values of the feature in one cluster. In level one, we have examined a single feature and in level two we have examined two features. First and the most important feature of the story was defined by the story ending (both levels feature multiple endings). The question then was: "Do stories grouped in one cluster end the same when using a particular input distance metric?" This is represented by *precision* of the metric. To obtain precision we first identified the most frequent ending within each cluster. Then we defined the cluster precision as the fraction of the stories with this ending inside the cluster. The resulting precision for input distance metric was then weighted average of precisions of all resulting clusters. Precision thus takes values between zero and one. Precision was used both in level one and in level two.

In level two we have also inspected how much time (in seconds) Thomas interacts with girls and how much time he stays alone. If the clustering works well with respect to this feature, we would expect stories in the same cluster to have similar values, e.g. Thomas interacts with Barbara in each story from one group roughly the same time. To measure this we define *time deviations* of a cluster that are the standard deviations of the respective times gathered from stories in that cluster. E.g. cluster with two stories where Thomas interacts with Barbara for 60 and 120 seconds has higher deviation than cluster with three stories where the respective times are 110, 120 and 130 seconds. The lower the deviation the more "similar" the stories are.

To summarize we have run k-means algorithm with four story distance metric on two domains with different resulting number of clusters multiple times. Story distance metric for k-means were tension curve distance and Levenshtein, Jaccard and Jaro-Winkler distance on action strings. We compared how similar are stories in resulting clusters according to two features – story ending and time the characters spend together in the story. As a baseline we used a random cluster assignment where the clusters were not determined by k-means but every story was assigned to a cluster at random – this randomization was repeated 8 times and results were averaged, precisely as with regular clustering runs.

4 Results

We have run the analysis on two datasets. First dataset was 1135 play sessions of game SimDate3D level one generated automatically with user input being simulated by a random algorithm. Second dataset was 41 human play sessions of game SimDate3D level two which exhibits more complex domain with more game endings.

4.1 Level One

From 1135 play sessions of game level one, 608 stories ended with characters getting angry with each other and parting, 479 ended with characters being too positive and not reaching the cinema, 16 stories ended with characters getting to the cinema and 32

stories endings were undefined (because of some technical issues). To compensate for having only 16 stories endings with characters getting to the cinema, we have used multi-sampling by a factor of 15 for these types of endings (meaning that each "cinema" story was present 15 times in the dataset). The low number of these stories is not surprising as it is necessary to "steer" the conversation between agents intelligently to assure they will get to the cinema – a thing that the randomized user input did not account for. Since there are three possible endings we clustered into 3 – 5 clusters.

Two level one stories with intimate ending:

"AABDBHHIMOIOIOJDFLCCDYHGLQPHPHPHPHPHPHPHPHPHPHPHLHQDFGIODFGIOODYGLHIDMH
NMNPHPHPHPHPHNHIOUUUUUVQWDP"

"AABDBHHMNIOIODFGLLCCPHPHPHPDMHNUUUUUVQWDP"

Two level one stories with breakup ending:

"AACPDMADPDDZG00123"

"AABDBHHMNIOMNJ6LCDMHMCMHNIOTLDTHMNMPHPHPHPHPHPHPMHNbDTTMZG00123"

Fig. 2. Level one stories action strings examples. The figure shows action strings of two stories with intimate ending and of two stories with breakup ending. Notice that the last few actions are the same for intimate endings and breakup endings (underlined).

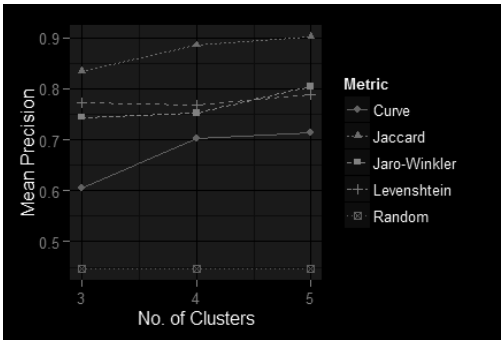

Fig. 3. Level one clustering results. Cluster precision weighted averages can be seen for three, four and five clusters. The results are averaged over eight clustering runs with different initial cluster positions.

Our results indicate that all of k-means distance metrics were able to cluster the simpler domain of level one with high cluster precision (Fig. 3). Quite surprising is the high value of precision for Jaccard index. Jaccard index compares only the number of same actions in both samples (their intersection divided by their union) not taking action positions into account. We believe that one of the causes of Jaccard index doing well is that two of the endings had pre-scripted final action sequences (Fig. 2) which biased the string distance metrics. For this reason, we removed all pre-scripted actions at the end of the stories for the experiments in level two. Other reason of string metrics doing well in the scenario in general is the relative simplicity – more negative actions in the scenario meant the probability of negative ending increased and vice versa. These straight-forward influences created a bias favoring the string

metrics in level one. The tension curve scored the worst with precision ranging from 0.6 for three clusters to 0.76 for five clusters. All metrics scored significantly better than the random cluster assignment.

4.2 Level Two

From 41 play sessions of level two, 14 stories ended with Thomas staying with Barbara, 21 stories ended with Thomas staying with Nataly, 4 stories ended with both girls breaking up with Thomas and 2 stories ended with Thomas staying in relationship with both girls. To compensate for the low number of endings of the latter two we have used multi-sampling by a factor of two for the stories with these endings. Since there are four possible endings we have tried clustering into 4 – 6 clusters. More clusters were not tried, because already with 6 clusters present, some of them were almost empty due to lower initial number of stories. Example action strings for level two are shown in Fig. 4.

Two level two stories with "stay with Barbara" ending:

"ABACDBDEF4APDQDNAOD1AkDEDrACA9D5DQDNAOAPACDEA2D2A2DQDNAPAOD1ACAKDKALDQDNAPDMACANDQDNAPAOACAPDQDNAOFQDNACAJAPDQDNAOFQDQF4D1"

"ABACDBDEF4DKAkDkAkDkAkDkDQDtAtAPABDBAGDEAPDQAGDEDlAlD1AID1AmDrAJD1A9FQFcF4"

Two level two stories with "stay with Nataly" ending:

"DBDEABACF4APDQDnAnDBABDEAqD1AkDkAkDkAkDIAIDIAIDIDQDhAMAPD1A9DQDhAqA2AgAPDQDhAgFQDQDhAgFBFQDQAqF4DEAPDL"

"F4ABACDBDEAmDJAkDQDtAtAPABDBAGDEAPDQAGDEDQDjAjAPABDBATDEAPDQATDEDJAmDmAlDlAlDlAlDlDQDhAMAPFQDQDhAgFBFQATDQAKF4"

Fig. 4. Level two stories action strings examples. Notice the sequences are longer than level one sequences. There are two reasons for this: a) level two takes longer to complete and thus more actions are performed and that b) in level two each action is coded by two letters, first letter marks the character that performed the action (D – Thomas, A – Barbara and F – Nataly), second letter marks the action, e.g. "AB" marks Barbara performing action "set focus".

The preliminary results on level two show a drop in cluster precision of string based metrics indicating the higher complexity of the domain. The tension curve metric demonstrated the highest precision. Without multisampling, the mean precision of curve metric for 5 clusters was 0.66, while random clustering had precision of 0.58. Multisampling has resulted in an increase of precision (0.73). See Fig. 5, upper left for overall trends with other distance metrics. The time deviations of all metrics were lower than that of random cluster assignment (see Fig. 5), however the gap is not large enough to allow for strong conclusions. Note however, that for "Nataly" and "Alone" time deviations the tension curve performs the worst, although only by a small margin. Still, no string distance metric performs better than tension curve considering the 4 tasks together.

The good performance of the tension curve may be an indicator that the tension curve scales better than the somewhat naïve string distance metrics. The problem of

the string distance metrics might be that they do not make any abstraction of the story from the qualitative point of view and may be misled by action sequences that look different, but are not different from the story perspective (e. g. talking about weather and talking about yesterday lunch both represent casual conversation but have different action sequences). Moreover, the action sequences contain actions that may not be directly related to story tension, e.g. move actions and set focus actions. One of the ideas how to improve the performance could be to prune actions like this from the sequence. However, note that results for level two are preliminary and the trends need to be confirmed by running the clustering on larger data sets and different domains.

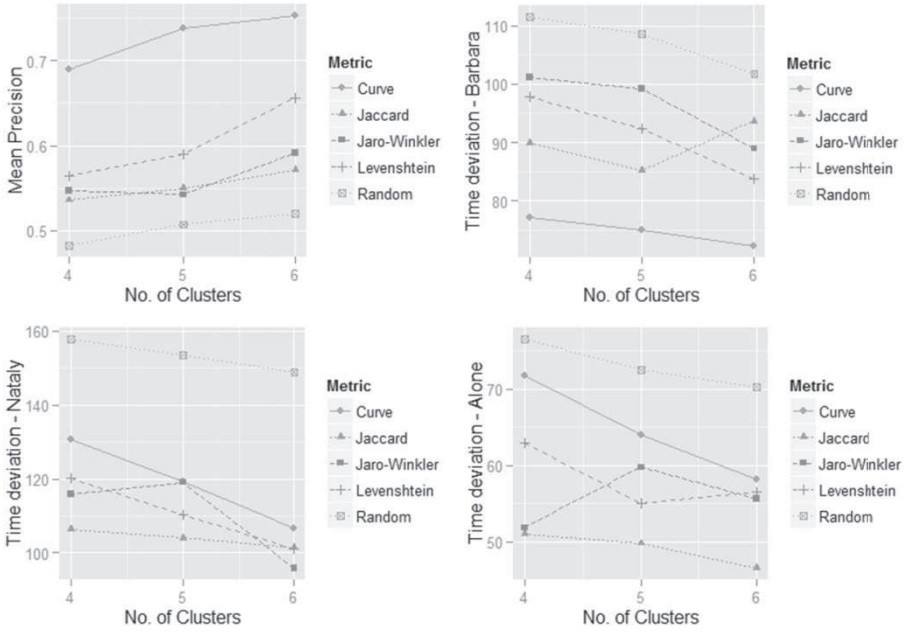

Fig. 5. Level two clustering results. Clusters precision plot (upper left) shows tension curve outperforming string metrics suggesting that it scales better. Time (in seconds) deviation plots for Barbara and Thomas interaction (upper right), Nataly and Thomas (lower left) and Thomas alone (lower right) show that all metrics are doing better than a random assignment (lower deviation means the stories are "closer" to each other in this sense). The results are averaged over eight clustering runs with different initial cluster positions.

5 Conclusion and Future Work

We have presented a methodology for semi-automatic evaluation of interactive storytelling systems based on a meaningful clustering of similar stories with k-means algorithm that could be plugged in an IS development cycle.

First results indicate that considering precision on a simple domain string distance metrics outperform tension curve metric (results from experiments on SimDate3D

level one), but they fail to scale well on a complex domain of SimDate3D level two, where tension curve metric outperformed them. This hints us that the tension curve metric is likely to scale better, but this needs to be confirmed by further experiments. The results for time deviations in level two are less clear and best metric could not be determined for that case.

The next steps are to evaluate this methodology on other domains and larger data sets. We are now in the process of gathering more data for SimDate3D level two.

As our future work we want to investigate the possibilities of using the tension curve to detect more fine-grained features of the input stories, e.g. conflict detection and classification. Other candidates for new features include separate evaluation of emotional state for each pair of characters. So far we have used k-means clustering with one distance metric at once, but k-means algorithm allows for combination of distance metrics. This could increase the clustering performance on stories.

More work could also be done to develop "smart" random user that would be able to explore parts of the story space more intensively, as instructed by the designer.

Acknowledgments. This work was partially supported by the student research grant GA UK 559813/2013/A-INF/MFF, by the SVV project number 267 314 and by the grant P103/10/1287 from GAČR.

References

1. Mateas, M., Stern, A.: Façade: An experiment in building a fully-realized interactive drama. In: Game Developer's Conference: Game Design Track (2003)
2. McCoy, J., Treanor, M., Samuel, B.: Prom Week: social physics as gameplay. In: Proceedings of the 6th International Conference on Foundations of Digital Games, pp. 319–321 (2011)
3. Aylett, R., Kriegel, M., Lim, M.: ORIENT: interactive agents for stage-based role-play. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, vol. 2, pp. 1371–1372 (2009)
4. Endrass, B., Rehm, M., André, E.: Planning Small Talk behavior with cultural influences for multiagent systems. Computer Speech & Language 25(2), 158–174 (2011)
5. Aylett, R., Vala, M., Sequeira, P., Paiva, A.: FearNot! – an emergent narrative approach to virtual dramas for anti-bullying education. In: Cavazza, M., Donikian, S. (eds.) ICVS 2007. LNCS, vol. 4871, pp. 202–205. Springer, Heidelberg (2007)
6. Cavazza, M., Lugrin, J., Pizzi, D., Charles, F.: Madame bovary on the holodeck: immersive interactive storytelling. In: Proceedings of the 15th International Conference on Multimedia, pp. 651–660 (2007)
7. Schoenau-Fog, H.: Hooked! – evaluating engagement as continuation desire in interactive narratives. In: Si, M., Thu, D., André, E., Lester, J., Tanenbaum, J., Zammitto, V. (eds.) ICIDS 2011. LNCS, vol. 7069, pp. 219–230. Springer, Heidelberg (2011)
8. Bída, M., Brom, C., Popelová, M.: To date or not to date? A minimalist affect-modulated control architecture for dating virtual characters. In: Vilhjálmsson, H.H., Kopp, S., Marsella, S., Thórisson, K.R. (eds.) IVA 2011. LNCS, vol. 6895, pp. 419–425. Springer, Heidelberg (2011)

9. Brewer, W., Lichtenstein, E.: Stories are to entertain: A structural-affect theory of stories. Journal of Pragmatics (1982)
10. y Pérez, P., Sharples, M.: MEXICA: A computer model of a cognitive account of creative writing. Journal of Experimental & Theoretical Artificial Intelligence 13(2), 119–139 (2001)
11. Ware, S.G., Young, R.M., Harrison, B., Roberts, D.L.: Four quantitative metrics describing narrative conflict. In: Oyarzun, D., Peinado, F., Young, R.M., Elizalde, A., Méndez, G. (eds.) ICIDS 2012. LNCS, vol. 7648, pp. 18–29. Springer, Heidelberg (2012)
12. Weyhrauch, P., Bates, J.: Guiding interactive drama. PhD. Thesis (1997)
13. Ontañón, S., Zhu, J.: On the role of domain knowledge in analogy-based story generation. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pp. 1717–1722 (2011)
14. Forbus, K., Gentner, D., Law, K.: MAC/FAC: A model of similarity-based retrieval. Cognitive Science 19(2), 141–205 (1995)
15. Cheong, Y., Jhala, A., Bae, B., Young, R.: Automatically generating summary visualizations from game logs. In: Proc. AIIDE, pp. 167–172 (2008)
16. Rabe, F., Wachsmuth, I.: An Event Metric and an Episode Metric for a Virtual Guide. In: Proceedings of the 5th International Conference on Agents and Artificial Intelligence, vol. 2, pp. 543–546 (2013)
17. Zwaan, R.A., Langston, M.C., Graesser, A.C.: The construction of situation models in narrative comprehension: An event-indexing model. Psychological Science 6(5), 292–297 (1995)
18. Porteous, J., Charles, F., Cavazza, M.: NetworkING: using character relationships for interactive narrative generation. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems, pp. 595–602. IFAAMAS (2013)
19. Aylett, R., Louchart, S.: If I were you - Double appraisal in affective agents. In: Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), pp. 1233–1236 (2008)
20. Kadlec, R., Čermák, M., Behan, Z., Brom, C.: Generating Corpora of Activities of Daily Living and towards Measuring the Corpora's Complexity. In: Dignum, F., Brom, C., Hindriks, K., Beer, M., Richards, D. (eds.) CAVE 2012. LNCS, vol. 7764, pp. 149–166. Springer, Heidelberg (2013)
21. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, New York (2001)
22. Freytag, G.: Technique of the Drama: An Exposition of Dramatic Composition and Art (1863)
23. Ortony, A., Clore, G.L., Collins, A.: The cognitive structure of emotions. Cambridge University Press, Cambridge (1988)
24. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
25. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles 37, 547–579 (1901) (in French)
26. Winkler, W.: String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In: Proceedings of the Section on Survey Research Methods (American Statistical Association), pp. 354–359 (1990)
27. Bída, M., Černý, M., Brom, C.: SimDate3D – Level Two. In: Koenitz, H., Sezen, T.I., Ferri, G., Haahr, M., Sezen, D., Çatak, G. (eds.) ICIDS 2013. LNCS, vol. 8230, pp. 128–131. Springer, Heidelberg (2013)