Follow-up on Automatic Story Clustering for Interactive Narrative Authoring

Michal Bída, Martin Černý and Cyril Brom[1]

Abstract. One of the challenges in designing storytelling systems is the evaluation of resulting narratives. As the story space is usually extremely large even for very short stories, it is often unfeasible to evaluate every story generated in the system by hand. To help the system designers to maintain control over the generated stories a general method for semi-automatic evaluation of narrative systems based on clustering of similar stories has been proposed. In this paper we report on further progress in this endeavor. We added new distance metrics and evaluated them on the same domain with additional data. We have also successfully applied the method to a very different domain. Further, we made first steps towards automatic story space exploration with a random user.

1 INTRODUCTION

Developing interactive storytelling (IS) systems is a challenging task involving multi-disciplinary knowledge, yet a number of IS systems was developed in the past, such as Façade [1], ORIENT [2] or FearNot! [3]. Bída et al. [4] notes that the evaluation of complex IS systems is a demanding process often requiring extensive effort. To mitigate this, the authors propose a computer assisted method of story evaluation based on clustering the stories into clusters according to their similarity. The general idea is that by meaningful clustering of the stories into groups the human designer will not be required to evaluate all the stories, but only few from each cluster and thus save development time. Authors also reported on the performance of the method on two domains - SimDate3D (SD) Level One and SD Level Two [5]. The first results indicated that the main metric could scale better than the other metrics on the complex domain of SD Level Two.

In this paper we report on further progress in a similar endeavor. Firstly, we have added two new features for the clustering algorithm in the SD domain - a) automatic extraction of sub-scenes from the recorded story and b) condensed tension difference curve based on the sub-scenes. We have managed to reproduce previous results on an extended domain of SD Level Two getting good performance using some of the new features. Secondly, we have implemented a random user that tries to explore the story space of SD Level Two by playing differently than an input set of previous stories hence exploring parts of story space not seen in the input set of stories. We show the performance of the metrics in distinguishing between stories generated by the random user and the original set of stories. Thirdly, we have applied the method on stories generated by the MOSS system [6] in order to investigate the performance of the method on a different domain.

Aside from the work mentioned, little has been done on story clustering. Weyhrauch [7] implemented several evaluation functions specific for his emergent narrative system. Ontañón and Zhu [8] proposed an analogy-based story generation system, where they evaluated the quality of resulting stories by measuring their similarity to “source” stories (input human-made stories). Compared to the approach in this paper, they were solving a problem of generation of the stories rather than the analysis of the stories.

This paper is organized as follows: First, we will describe the story domains we used in the experiments, then we will discuss updates of the method for narrative analysis and afterwards we present results of the new experiments. We will conclude the paper with discussion and future work.

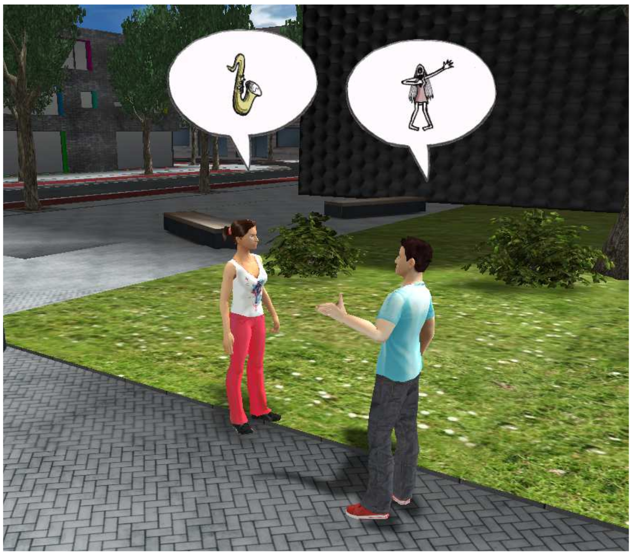

Figure 1. SimDate3D Level Two screenshot showing Thomas and Nataly in the park with emoticons above their heads having a conversation about music.

2 DOMAINS

The experiments detailed in this paper have been conducted on IS system SD Level Two detailed in [5] and MOSS system [6].

SD game (Figure 1) is a 3D dating game taking place in a virtual city, with three protagonists: Thomas, Barbara and Nataly. The characters communicate through comic-like bubbles with emoticons indicating the general topic of the conversation

[1] Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic.
Email: {michal.bida,cerny.m}@gmail.com, brom@ksvi.mff.cuni.cz

(see Figure 1). The user partially controls one of the characters actions (typically Thomas). The users' goal is to gain the highest score by achieving certain kind of things, e.g. Thomas kissing one of the girls. The game features four possible endings.

The MOSS system [6] developed by M. Sarlej generates short stories with morals (e.g. greed, retribution, etc) in three domains (animals, family and fairytale). Each moral has its own emotional pattern that is used to generate stories with moral of a particular category. Internally the system uses Prolog abstraction to generate the stories, which is then translated to human readable text with Perl scripts. We worked directly with the internal prolog representation of the stories, which we parsed and analyzed with the system.

3 METHOD

Here, we will briefly overview the method we evaluated (which is given in detail in [4]). The main idea is to cluster the resulting narratives of a given IS system into groups of similar stories. The human designer then needs to see only several stories from each group to gain sufficient understanding of all the stories the cluster contains, saving development time. The clustering is done with the k-means algorithm. In the previous work, the clustering was based on two general features of stories: a) story action sequence and b) story tension (dramatic) curve.

The story action sequence is created by taking the sequence of actions done by all the characters in the story. Each of the actions available in the domain is assigned a letter and the sequence of these letters forms the *action string*. This way, standard string distance metrics (*Levenshtein*, *Jaro-Winkler* and *Jaccard* distances) are applicable to measure similarity between *action strings* representing different stories. In previous work, Jaccard distance has been shown to be of little use for story clustering in SD domains and is therefore tested here only for MOSS stories.

The *tension curve* is extracted from emotions experienced by the story protagonists. In SD this is straightforward as the characters are equipped with emotion model. The tension in SD is computed as follows: Every 250 ms we make a snapshot of all characters' emotions. Then we take the sum of these emotions where every positive emotion is counted with a minus sign and every negative emotion is counted with a plus sign. The resulting number encodes the tension value at the moment. The *tension curve* is then simply the piecewise linear function defined by these values.

In the MOSS system the emotions are also defined explicitly as a part of the generated stories. We again take the sum of positive and negative emotions at each time point of the story and the resulting value is the tension value at the specific time point of the story.

We propose two new features for clustering the SD stories: *sub-scene sequence string* and *condensed tension difference curve*. A sub-scene is a time span in the story where a) the set of characters that are in the proximity of the main protagonist do not change and b) the location of the main protagonist does not change. Let us suppose that Thomas (the main protagonist) is with Barbara (character) at the restaurant (place) – this is one sub-scene. After 5 minutes, Nataly arrives and joins them. At this moment, the old sub-scene ends and a new one begins. The new sub-scene features Thomas, Barbara and Nataly at the restaurant. Sub-scenes are extracted automatically from the story logs. The time span of sub-scenes varies from 5 seconds (enforced lower limit) to the whole duration of the story.

To measure distance between sub-scene sequences we assign strings to sub-scenes in the following way: one letter represents a location of the story (e.g. P for park) and the consecutive letters represent characters in the sub-scene (e.g. T for Thomas; one letter per each character present). For example, the "TBR" string represents a sub-scene where Thomas is with Barbara at the restaurant. The sub-scene sequence string is simply a concatenation of the individual strings. We then apply string distance algorithms as is the case with action strings.

Condensed tension difference curve is extracted from sub-scenes. We look at the tension value at the beginning and at the end of the sub-scene. The difference between these two values represents the tension difference for respective sub-scene. The condensed tension difference curve is defined as a sequence of all of these differences.

We have not implemented sub-scenes for MOSS stories, because the MOSS stories are already relatively short and composed of at most two sub-scenes. To check whether the clustering really captures non-trivial properties of the stories, we also tested difference in story length as distance metric for the MOSS domain.

All pairwise distances between stories have been computed, normalized and standardized prior to clustering.

3.1 Story space exploration with a random user

IS systems are often interactive, requiring a human user in the loop. Exploring the story space of such systems may be problematic as one needs many users and many story runs to get a reasonable coverage of the story space. For semi-automatic analysis the designer would benefit from an algorithm that would be able to explore parts of the story space automatically. For SD we have implemented a random user that is able to play the game alone. In addition, the random user tries to steer away from a given set of stories. Hence exploring parts of story space not covered in the given set of stories revealing previously unseen parts of the story space to the designer. This is achieved as follows: The random user (controlling Thomas) extracts the sub-scene sequences from the given set of stories and then tries to achieve a different sub-scene sequence in the story he is playing in. E.g., if the random users detects that most of the given stories started with characters at the restaurant, he will try to change location in the story by inviting the characters for example to the cinema and so forth for the second and the n-th sub-scene in the sequence. The random user has simple domain-specific knowledge that limits the actions he considers only to those contextually appropriate (e.g. he does no try to become intimate with a girl at the restaurant).

3.2 Evaluating clustering quality

As there is no generally accepted method for evaluating the quality of a clustering independent of the application, we use ad hoc method suitable for our scenario. Intuitively, a clustering is good, if stories in the same cluster have many features in common. Let us have a feature function $f: S \rightarrow V$, where S is the set of all possible stories and V is a finite set representing possible values of a feature the designer might be interested in.

For a cluster $X \subset S$ we define *precision with respect to f* as the proportional size of its largest subset sharing the same value of the feature:

$$precision(X, f) = \frac{\max\{|M| : M \subset X, \forall m, n \in M : f(m) = f(n)\}}{|X|}$$

In other words, precision of 0.62 means 62% of stories in the cluster produce the same value for *f*. The precision of the whole clustering is simply the average of per-cluster precisions. A system that clusters stories can be considered useful, if it provides high precision across multiple domains and multiple features.

In the experiments, we tested three features: the ending of the story (Experiment 1), the type of user (random vs. human) that generated the story (Experiment 2) and the MOSS moral of the story (Experiment 3).

As k-means depends on random initialization we ran each analysis 100 times to get robust results. In further text, we always report the average precision of these 100 clustering runs. To provide a simple baseline to the measurements, we also tried assigning stories to clusters at random. Once again an average of 100 random assignments is measured.

To provide a more robust evaluation of the methodology, it would be best to measure precision with respect to similarity of stories as perceived by humans. This however poses multiple methodological issues. In our view, a biggest obstacle to human evaluation is finding a useful dataset. Since humans cannot effectively cluster more than a handful of stories, the dataset needs to be small, which is usually unsuitable for machine clustering as the algorithm can easily pickup artifacts in the data. We left this as a future work.

4 EXPERIMENT 1

In Experiment 1 we analyzed an extended dataset of 70 human play sessions of SD Level Two using additional features – sub-scenes sequence string distance and condensed tension difference curve based on sub-scenes. Precision is measured with respect to the ending of the story. A graph of the results is presented in Figure 2.

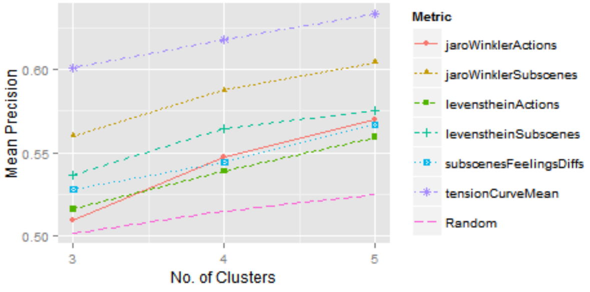

Figure 2. SD Level Two clustering results. Cluster precision weighted averages can be seen for three, four and five clusters (this is chosen arbitrarily based on that there are four possible endings). The results are averaged over 100 clustering runs with different initial cluster positions. The precision is calculated with respect to story ending.

As in previous work [4] we see that the tension curve outperforms other approaches in mean precision (0.6 for three clusters to 0.63 for five clusters). The interesting observation is that the sub-scene string sequence (metrics marked as “Subscenes” on Figures 2, 3, 4) outperform action strings (metrics marked as “Actions” on Figures 2, 3, 4) on this dataset. This indicates that sub-scene sequence is a meaningful feature in SD domain, relevant to story ending. Also note that Jaro-Winkler distance on sub-scenes (average 0.58) slightly outperforms Levenshtein (average 0.56). This is somewhat unexpected as Jaro-Winkler distance is usually a sub-par choice for clustering as it does not satisfy the triangle inequality. However this distance gives more weight to differences between first four characters of the string. The good performance of Jaro-Winkler on sub-scene sequences may then be explained by a large impact of the beginning of the story on its ending. Assigning higher weight to story start and/or story end might be an interesting extension of the approach as it would reflect the way stories are perceived by humans.

The compressed tension difference curve (metrics marked as “subscenesFeelingDiffs” on Figures 2, 3) scored on par with action strings distance metrics (average 0.55), but did not match the uncompressed original tension curve.

All metrics scored significantly better than the random cluster assignment. However compared to previous results [4] the addition of more stories resulted in lower precision for all previously measured metrics (tension curve and action strings). This might be partly caused by the larger size of the dataset, but it indicates that the metrics need to be made more robust.

Examples of the stories from this dataset and their clustering can be found in the appendix.

5 EXPERIMENT 2

In Experiment 2 we analyze a dataset containing 41 original human play sessions (as analyzed in [4]) and 66 randomly selected play sessions gathered from the random user. Precision is measured with respect to the type of the user that generated the story. A graph of the results is presented in Figure 3.

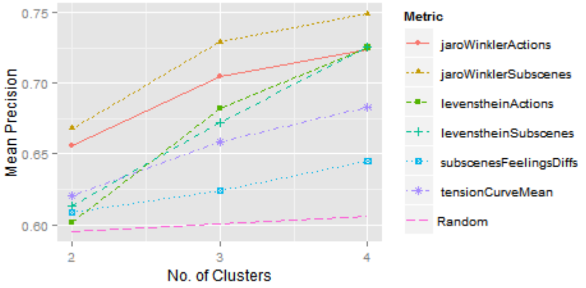

Figure 3. Experiment 2 clustering results. Figure shows the average precision of clustering with respect to the users that created the stories as a function of number of clusters.

The best metric for distinguishing between human and random user is Jaro-Winkler distance on sub-scenes (with precision 0.67 on two and 0.72 on four clusters). This can be explained again by the feature of the algorithm putting more weight on the first characters of the string. The random user tried to achieve different sub-scene sequence than the human users. Even though the story always begins the same (the first sub-scene is always the same), the random user immediately tried to change the sub-scene, so the second one differed from the average done by human users. This was picked up by Jaro-Winkler resulting in better performance of the algorithm.

The tension curve performed worse on this task (average 0.65). This is understandable as different sub-scene sequences in the story may produce similar tension curves. However this also indicates that the problem of similarity of the stories is multi-layered and to grasp this properly a combination of features is likely to be required.

6 EXPERIMENT 3

In Experiment 3, we ran the method on stories generated by the MOSS system. We have analyzed 3000 stories from fairytale domain of MOSS with recklessness, retribution and reward morals (1000 from each). Half of the stories comprised of two dramatic actions, and the other half comprised of four dramatic actions. In both cases, the resulting stories contained about 30 atomic actions. The precision was measured with respect to the moral of the story. A graph of the results is presented in Figure 4.

We can see that the precision of clustering is very high for almost all clustering metrics. For MOSS stories of length four, tension curve achieved precision of 0.99 on three clusters. The sum of normalized story length and Levenshtein on action strings was the second best scoring 0.93 on three clusters. On MOSS stories with length two, these two metrics performed a bit worse. The best was Levenshtein on action strings which averaged on 0.94 and the tension curve with 0.88 precision on average. The story length metric was outperformed by almost all other metrics and it also did not bring significant improvements to the Levenshtein distance indicating that the MOSS generating process did not produce artifacts in story length. Similarly to previous results on the SD domain, Jaccard distance did not perform well.

This overall good performance is caused by the fact that stories in MOSS are generated through templates that use emotional patterns. Stories in one domain exhibit the same or very similar emotional patterns resulting in similar tension curves. This is picked by the tension curve metric really well. The comparable performance of string metrics on action strings is likely caused by the presence of emotional actions in the action strings. The overall slightly worse performance on stories with dramatic length two is probably caused by the fact that less dramatic actions in the story offer less space to distinguish the stories from each other (however the performance was still remarkably good).

Examples of the stories from this dataset and their clustering can be found in the appendix.

7 CONCLUSIONS AND FUTURE WORK

We have presented new data for a methodology for semi-automatic evaluation of interactive storytelling systems based on clustering of similar stories. We have reproduced and refined previous results in the area.

New results showed that the method can be transferred successfully to other domain. However we need to take this with a grain of salt as the MOSS story generator abstraction was very favorable to the method as it uses emotional patterns to define categories of the stories.

Next, we have added new feature of stories, sub-scene sequence, that was used in the implementation of random user designed to explore unvisited parts of the story space of SimDate3D domain and we have shown the performance of the method on distinguishing random user from the human users. Some of the metrics scored worse than expected indicating that to grasp story similarity properly a combination of features will be required.

The semi-automatic exploration of the story space with a random user proved useful and will be further investigated in future work.

We have also shown the performance of the method on an extended dataset from SimDate3D Level Two. Although we

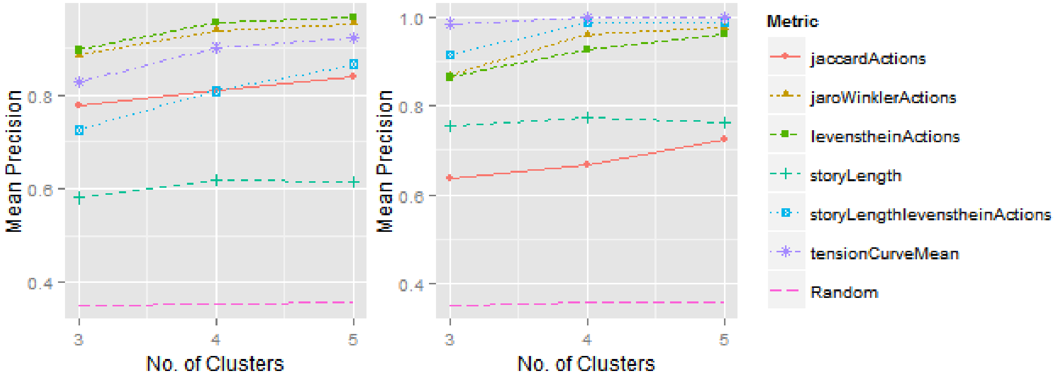

Figure 4. Experiment 3 MOSS domain clustering results. On the left there are precisions of clustering for three, four and five clusters when distinguishing between stories of the dramatic length two with particular moral. On the right there is the same for stories with the dramatic length four. All results were averaged over 100 clustering runs.

have reproduced the performance ordering of the metrics, the overall results were worse than in previous paper. The reason may be that the metrics do not accurately represent story similarity and pick a large amount of noise. A detailed analysis of stories in the same clusters could shed more light onto this and it is planned as future work, including comparison with story similarity as perceived by humans.

In line with conclusions from previous work, the tension curve provided best overall results across domains and feature functions, but as it did not work very well in Experiment 2 it cannot be considered universal and better metrics are needed. A combination of tension curve and one of the string distances might prove useful.

Other future work includes experiments with combination of distance metrics for the clustering algorithm and further enhancements and additional experiments with the random user. Finally, it would be beneficial to experimentally determine, how humans would cluster some of the stories.

ACKNOWLEDGEMENTS

This research is partially supported by SVV project number 260 224 and by student grant GA UK No. 559813/2013/A-INF/MFF.

REFERENCES

[1] M. Mateas, and A. Stern. Façade: An Experiment in Building a Fully-Realized Interactive Drama. In: *Game Developer's Conference: Game Design Track.* (2003).

[2] R. Aylett, M. Kriegel, and M. Lim: 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).

[3] R. Aylett, M. Vala, P. Sequeira and A. Paiva: FearNot! – An Emergent Narrative Approach to Virtual Dramas for Anti-Bullying Education. In: *Virtual Storytelling. Using Virtual Reality Technologies for Storytelling*, LNCS Vol. 4871, Springer, pp. 199-202, (2007).

[4] M. Bída, M. Černý, C. Brom: Towards Automatic Story Clustering for Interactive Narrative Authoring. In: *Interactive Storytelling.* LNCS, Vol. 8230, Springer, pp. 95–106. (2013)

[5] M. Bída, M. Černý and C. Brom: SimDate3D – Level Two. In: *Proceedings of ICIDS 2013.* LNCS, Vol. 8230, Springer, pp. 128-131. (2013)

[6] M. Sarlej, M. Ryan.: Generating Stories with Morals. In: *Interactive Storytelling.* LNCS, vol. 8230, Springer, pp. 217-222. (2013)

[7] P. Weyhrauch: Guiding Interactive Drama. PhD. Thesis, Carnegie Mellon University, Pittsburgh (1997).

[8] S. Ontañón, and J. Zhu: 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).

APPENDIX – EXAMPLE STORIES

Here, we present several examples of stories from SimDate3D and MOSS domains and show examples of the clustering of stories using the tension curve metric. In both cases we provide simple handcrafted natural language representations of the actions in the story.

A. SimDate3D Domain

Story 1: Thomas went with Barbara to the cinema. After the movie, he was rude to her. They have parted ways. Thomas went to Nataly's home to pick her up. They went out for a walk, but they did not speak much. Thomas insulted Nataly. They met Barbara. An argument started and both girls left Thomas.

Story 2: After the movie, he was rude to her. They have parted ways. Thomas went to Nataly's home to pick her up. Thomas was rude to Nataly. They went out for a walk and Thomas was rude to Nataly. They met Barbara. An argument started and both girls left Thomas.

Story 3: Thomas spent a long time with Barbara in the cinema, then he was very rude on her. Nataly was in the restaurant alone. Then Thomas and Barbara got very angry on each other, but continued talking. Nataly noticed them on their way from restaurant and she run towards them. An argument started and Thomas ended up with Nataly.

Stories 1, 2 and 3 get clustered together in most cases. Stories 1 and 2 are extremely similar and end the same, while story 3 is an example of a story that is relatively similar to the other two, but does not end the same.

Story 4: Thomas and Barbara were on a way to cinema. Thomas asked Barbara to kiss him and to cuddle, she refused. Then they've run into Nataly, argument started and Thomas ended up with Barbara.

Story 5: Thomas and Barbara were going to the cinema. Thomas was making jokes on the way. Before they've get to the cinema they've run into Nataly, argument started and Thomas ended up with Barbara.

Stories 4 and 5 on the other hand are also very similar and end the same but were almost never clustered together.

B. MOSS Domain

Story 1: A wizard gets hungry. He picks up a rose. A troll kidnaps a princess. The troll also kidnaps a dwarf. A knight rescues the princess from the troll. (Generated as an example for recklessness)

Story 2: A wizard gets hungry. He picks up a rose. A troll kidnaps a princess. The troll also kidnaps a dwarf. A dragon gives a treasure to the dwarf. (Generated as an example for recklessness)

Story 3: A dwarf kills a princess. A troll kidnaps the dwarf. A dragon tries to kidnap a unicorn, but fails. Fairy gives magical dust to the dragon. Dragon gives the dust back to the fairy. (Generated as an example for retribution)

All those stories are from the same cluster. While it is clearly visible, how stories 1 and 2 are extremely similar, story 3 seems very different.