Experimental Results from a Rational Reconstruction of MINSTREL

Brandon Tearse Peter Mawhorter Michael Mateas Noah Wardrip-Fruin
Department of Computer Science
University of California, Santa Cruz
Santa Cruz, CA 95064
{batman, pmawhorter, michaelm, nwf }@soe.ucsc.edu

Abstract

This paper presents results from a rational reconstruction project that is aimed at exploring the creative potential of Scott Turner's 1993 MINSTREL system. In particular, we investigate the properties of Turner's original system of Transform-Recall-Adapt Methods (TRAMs) and analyze the performance of an alternate TRAM application strategy. In order to estimate the creativity of our reconstructed system under various conditions, we measure both the variance of the output space and the ratio of sensible to nonsensical results (as determined by hand-labeling). Together, these metrics give us insight into the creativity of the algorithm as originally constructed, and allow us to measure the changes that our modifications induce.

Introduction

Scott Turner's 1993 MINSTREL system (Turner 1994) is considered a high water mark for computational story generation. This system was based on the notion of imaginative recall: the idea that stories can be built by adapting fragments from other stories and jigsawing them together into something new. Turner argued that MINSTREL was a computer simulation of human creativity and that the Transform-Recall-Adapt Methods (TRAMs) were the cornerstone of that creativity, but never fully evaluated his system.

This paper formally evaluates the creativity of MINSTREL's TRAM system. Previous work done by Tearse et al. discussed an initial reimplementation of MINSTREL (Tearse, Mateas, and Wardrip-Fruin 2010) and in this paper we detail our experiments which seek to shed light on the creativity exhibited by TRAMs in MINSTREL REMIXED. We have continued the rational reconstruction efforts on the system and have attempted to measure creativity according to Turner's original definition. Turner states that "Whether or not something is creative depends on the number and quality of its differences from similar works," (Turner 1994) so our goal is to measure the variety and quality of the TRAM system's output relative to its input (the stories that it knows). Specifically, we measure the expected amount of variance between results, and whether they are sensible or nonsensical. Although the diversity of results is not a direct measure of variance relative to similar works, as the diversity increases, the likelihood that the system generates more creative results does as well, so diversity of the results (along with their quality) is an appropriate proxy for the creativity of the system. Beyond measuring the creativity of the original system, we also modify the system in order to shed light on the trade-offs that it gives rise to between quality and diversity. In running these experiments, we aim to validate the performance of MINSTREL as a creative system relative to Turner's definition of creativity.

Related Work

Rational Reconstruction

Rational reconstruction is done to investigate the inner workings of a system, ideally identifying the differences between implementation details and core processes. Projects such as Musen et al. (Musen, Gennari, and Wong 1995) have successfully used rational reconstruction to better understand the fundamental concepts of their system. A partial reconstruction of MINSTREL was even performed (Peinado and Gervas 2006) in which the knowledge representation systems of MINSTREL were recreated in W3C's OWL. While this did a good job of proving that the knowledge representation can be successfully recast, without the full system in place it could not be used to investigate other aspects of MINSTREL.

CBR-Based Storytelling Systems

Case Based Reasoning (CBR) is a popular approach for creating intelligences and is what MINSTREL's TRAM searching system is based on. While most CBR systems try to match input to a library and then adapt the responses into useful results, MINSTREL goes a step further by transforming its query in order to locate matches which are further afield which in turn increases its creative options. Other systems have used enhanced CBR to develop stories or character actions (Fairclough and Cunningham 2003; Gervas et al. 2005). While these systems are all interesting to look at, Turner in particular made claims about his system's creative output which can now be investigated.

Creativity

The idea of computational creativity is an area of interest for computer science, in part because it is closely tied with the notion of artificial intelligence. Unfortunately, creativity is difficult to define and although Boden and Ritchie et al. have

both presented guidelines and measurements (Boden 2004; Ritchie 2007), their concepts remain difficult to implement. Given the nature of our specific system and the general agreement between Boden, Ritchie, and Turner on the importance of variety and quality, we decided to measure creativity based on Turner's (Turner 1994) suggestion, looking at the variety and quality of the results. By using Turner's original definition, we are also measuring the system by its own standards, so to speak: our results will have direct bearing on whether Turner's system should be viewed as a success on his terms. Although measuring the number of possible outputs and the quality of those outputs (by measuring both size and sense to nonsense ratios) is useful, we also found it interesting to cluster the results in a manner similar to Smith (Smith et al. 2011) in order to get a better picture of actual result differences.

Method

Rational Reconstruction

As a rational reconstruction, the ultimate goal of our project is twofold: recreate MINSTREL in a form that can be used by others and investigate the design choices that went into MINSTREL in order to learn about the creative potential of the system. To achieve these goals, we have created MINSTREL REMIXED, which consists of a Scala codebase that reimplements the functionality of the original MINSTREL. Working from Turner's 1993 dissertation, we have tried to create a faithful reproduction of the original while introducing modularity. This modularity has allowed us to explore alternatives to several of Turner's design choices, thus better characterizing the tradeoffs faced by the system.

TRAMs

The TRAM system performs all of the fine-grained editing in MINSTREL REMIXED. TRAMs are small bundles of operations designed to help recall information from the story library. TRAMs are used to return information by giving them a query (a graph containing nodes describing some story fragment, using MINSTREL's graph-based story representation). The TRAM transforms the query, finds matches in the story library, and adapts one of those matches before returning it. Of course, the process of finding matches in the library may require further use of TRAMs. The TRAM system has the task of choosing which TRAMs to use during this recursive querying process.

An example of TRAMs in action using one of Turner's original King Arthur stories is as follows: a graph is passed in requiring a knight, John, to die by the sword. By transforming this query using a TRAM called Generalize Constraint, we might end up with a query in which something dies in a fight. If this then matched a story about another knight, Frances, who kills an Ogre, the TRAM system could replace the Ogre with John the knight and return a fragment about a duel between John and Frances in which John dies.

The creative power of TRAMs ultimately comes from their ability to find cases in the story library which aren't easily recognizable as applicable. In the original MINSTREL, queries that fail to return results are transformed and resubmitted. This leads to a random sequence of transformations before a result is located. In MINSTREL REMIXED however, we have implemented a weighted random TRAM selection scheme. This allows both the original functionality and more targeted weight based TRAM application to be used. The targeted TRAM approach results in fewer alterations being needed to get results and thus more similarity between the original query and the eventual result.

To provide a concrete illustration of the TRAM system and its ability to produce varying results, we can look at other ways for John the knight to die. If we start with a fragment in which John is required to die and then alter it with one TRAM to require an ambiguous health change rather than death and then another which replaced John with a generic person, the resulting query matches with the story fragment from figure 1. Generic John's health changing action matches to Princess Peach who uses a potion to hurt (rather than kill) herself. Upon adaptation back to the original requirements, the resulting fragment is that John commits suicide, killing himself with a potion (Figure 1 shows the process of generalization, recall, and adaption from left to right).

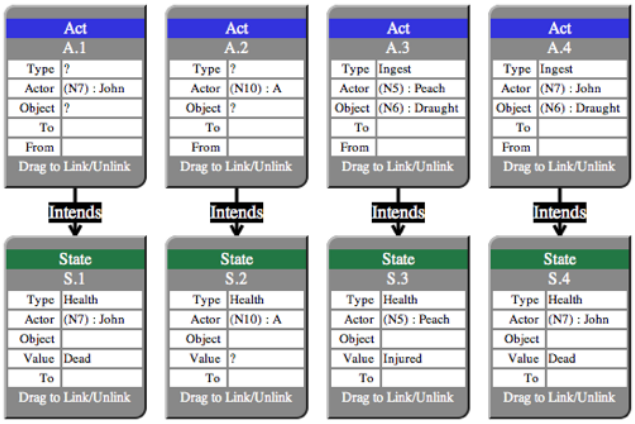

Figure 1: An example of the transform and adaption process.

Measuring Creativity

Many notions of creativity focus on the differences between system input and output. In particular, a system can be considered creative when it produces outputs that are significantly different from the given input (Pease and Winterstein 2001). Of course, the ability to generate a great variety of nonsensical outputs is not particularly impressive: variety must be balanced against quality. Unfortunately, both story quality and "significant difference" are difficult to measure in the realm of stories. In domains that have less dense semantic information, perceived difference between artifacts is usually related to measurable qualities of those artifacts (for example, pitch or duration in music are directly measurable and affect the perception of a piece). Given measurable qualities that relate to difference, a distance function can be used to cluster the output of a generator, and the resulting clustering will characterize how much meaningful variety

that generator can produce. For stories, however, significant difference is difficult to measure, let alone the task of creating a distance metric between stories. Measurable qualities, like word differences or story length, have unpredictable influences on the difference between two stories (for example, the same story could be told twice with very different overall lengths). More generally, every measurable aspect of a story has some relevance to the story, but changing less-relevant aspects of the story can result in insignificant differences. The problem of deciding which details of a story are relevant enough is itself a difficult unsolved AI problem; creating a computational distance metric over stories that corresponds to human perceptions would be a difficult task.

Besides characterizing the variety of our output, we hope to measure its quality. Ideally, each generated story could be assigned a quality value, and then a result set could be evaluated in terms of the quality of the results that it contained. Of course, computationally evaluating story quality is also an open problem, so for both variety and quality we are forced to rely on estimates. To estimate the overall quality of a result set, we hand-label each distinct result in the set as either sensible or nonsensical. By using a binary labelling, we come up with a relatively concrete metric over our set of results (an integral scale would be subject to more noise during labelling and would require a subjective weighting function to be comparable between tests).

To estimate the variety of our result space, we measure the expected variation between a pair of results, and use that as an estimate of the rate at which novel artifacts will be generated. The higher this estimate, the greater variety of artifacts we expect will occur for a fixed-size result set. And although we do not know exactly how variety among generated artifacts (measured using differences at the symbolic level e.g. a princess is used instead of a knight) affects the variety of the results in terms of significant differences, we can assume that they are correlated (i.e. the more varied the results in terms of raw symbol differences, the more likely it is that they contain stories that are significantly different).

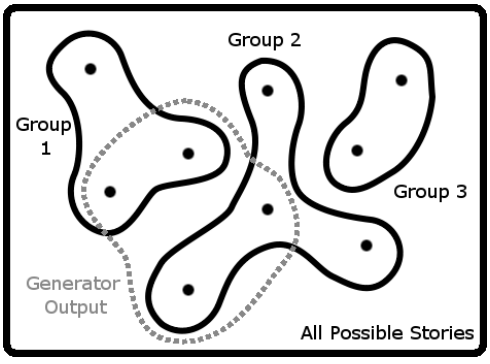

Figure 2: The story space divided into creative groups.

Figure 2 is a demonstration of our estimation method for result set variance. Within the entire space (all possible stories), each point represents a particular story (along with minor variants of that story, such as stories that substitute one particular weapon for another in a fight scene). These points in turn are grouped into sets each of which is composed of stories that are perceptually similar (so significant differences exist between these story groups, but not within them). These actual groupings are unknown (because automatically measuring significant difference between stories isn't yet possible), but we can measure the number of different particular stories (individual points) that our generator can be expected to produce. Although the exact correspondence is unknown, it is clear that the greater the variety of particular stories our system generates, the more story categories it can be expected to create examples of (in this case, if our variance measure were to increase from four to five by adding a random story, there would be a two in five chance that that increase would also increase the creativity of the system by including a story from group 3). So our measure of story variance is a proxy for the number of creative categories that our system will produce examples of, which in turn is (along with our measure of the sensibility of the results) a measure of the creativity of the system.

Experimental Setup

To measure the creativity of MINSTREL's TRAMs, we performed a variety of experiments each of which involved a single query. For each experimental condition, we ran five runs of 1000 repeated queries. We then calculated averages for each condition and ran four metrics on each. First we tallied the number of unique results (by collapsing exact duplicates). Next we computed the number of sensible versus nonsensical results using a mapping from results to sensibility that we built by hand which covered all results. Finally, we computed both the probability that a pair of queries under the given experimental conditions would have at least one difference, and the expected number of differences between such a pair. In addition to these measures we compute separate sense to nonsense ratios (s/ns) for just the unique results. Using these numbers, we are able to characterize the creativity of MINSTREL under our various conditions by comparing them to the baseline.

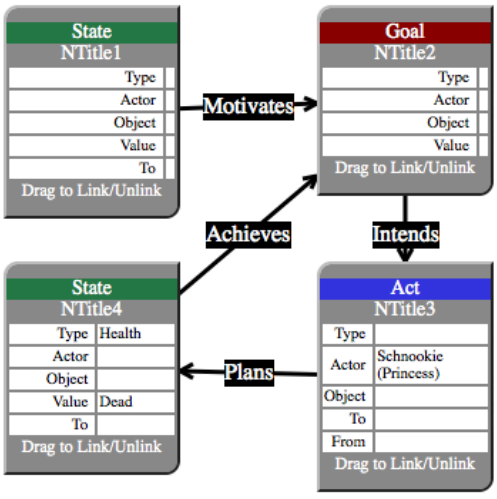

Figure 3: Our default query.

Each experiment varied one of five underlying variables: the depth limit used to find results (how far TRAMs can recurse), the query supplied, the story library used, the set of TRAMs available, and the weights on the TRAMs in use. For the base case, we used a search depth of five, a default test query, our full set of story libraries, our full set of TRAMs, and uniform TRAM weights. Our default query consisted of an unspecified State node connected to a Goal node, which connected to an Act node and then another State that linked back to the Goal. In terms of links, the first state motivated the goal, which planned the act, which intended the second state, which achieved the goal (Figure 3 shows this structure). In the Act node, we specified "Shnookie the Princess" as the actor, and in the second State node, we specified a type of "Health" and a value of "Dead". Beyond these constraints, the nodes were completely unspecified, so our query corresponded to the instruction: "Tell me a story in which Shnookie the princess kills something." Our story library contained sixteen hand-authored stories which ranged from simple ("PrincessAndPotion", in which a princess drinks a potion and becomes injured), to fairly complex (Our largest story contained 26 nodes and included eight different nouns). On average, our stories contain 11.5 nodes (Goals, Acts, States, and Beliefs) and include 4 nouns.

For these tests, we used a limited set of TRAMs that focused on Act nodes (hence the structure of our query). To test the full range of TRAMs, we would need to engage MINSTREL's author-level planning system in order to perform multiple lookups over a single query, which would make it difficult to make statements about the TRAM system in isolation. Given our limited query, we use a total of seven TRAMs: GeneralizeConstraint, GeneralizeActor, GeneralizeRole, LimitedRecall, RecallActs, IntentionSwitch and SimilarOutcomes.

Results

To get a sense of the TRAM system's creativity, we can look at our baseline result. We find that our measure of expected variance is about 7 (6.90), while our measure of quality is near 0.5 (0.544). In other words, if we were to submit two queries using these parameters, we'd expect about seven fields in which the results would differ, and only half as many sensible results as nonsensical ones on average. These parameters are not desirable for use in producing stories, because they produce too many nonsensical results (only about 35% (35.2%) of total results were sensible) but for testing the system, the even mix of sense and nonsense allows us to observe how changes promote one or the other. The expectation of seven differences between two random results is encouraging: it indicates that there is variety in the output space. Looking at the total number of stories, we can see that 1000 trials produces an average of about 70 (68.8) unique stories, about 21 (21.0) of which will be sensible. Among unique results, the s/ns ratio is around 0.4 (.439). The fact that this ratio is higher among total results than among unique results indicates that the sampling of unique results is biased towards sensible ones: sensible stories are repeated more often than nonsensical ones. Our baseline sets a high bar for variance, but exhibits lackluster quality.

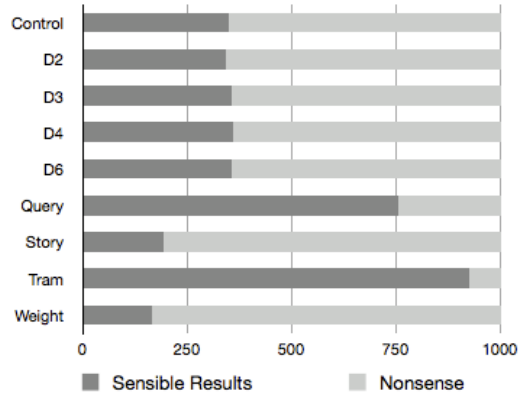

Figure 4: Total sensible versus nonsensical results.

Looking at the results as a whole, we can see some significant differences between the various cases. In figures 4 and 5, Our baseline initial query is shown under the heading Control. The D2, D3, D4, and D6 headings are for TRAM depth limits of 2, 3, 4, and 6 respectively (the initial query and all others used depth 5). In terms of the proportion of total results which are sensible, the depth doesn't seem to make a difference. Additionally, depths 3, 4, 5, and 6 are all roughly equivalent in terms of the total number of responses generated, both sensible and nonsensical. Depth 2 does generate significantly fewer results, however, although the s/ns ratio is still approximately the same as it is in the other runs. We can hypothesize that although a significant number of possible stories require at least three TRAMs to reach from our test query, the distribution of these depth-3 stories in terms of sense and nonsense is roughly the same as the distribution of results at depth 2. Based on this idea, we graphed the actual distribution of results across TRAM depth for the D6 test (shown in figure 7, which also shows results for the Weight case). This confirmed our hypothesis: most results have depth 2, many others have depth 1 or 3, but after depth 3, the number of results falls off sharply. Looking at just unique results (figure 8), we can see that there is an even more marked bias towards lower depths, with extremely few results at depths 5 and 6. Comparing figures 7 and 8 we see that many of the results at depths 5 and 6 are almost certainly identical to results at lower depths, unless the deeper unique results are repeated much more often. Examining the log files in detail confirms this. In terms of the MINSTREL system then, it appears that some TRAMs might have no effect on the result of a search, possibly because other TRAMs later reverse their effects. Of course, these TRAMs do have an impact on the overall distribution of results, even if they sometimes don't effect individual searches, and the TRAMs are only sometimes ineffectual.

After experimenting with the TRAM depth, we ran an alternate query to explore how much our results might depend on the query. The alternate query had the same graph structure, but stipulated that the goal and motivating state nodes had type 'Health', and that the actor of the act was a troll rather than a princess. Additionally, the state node in the

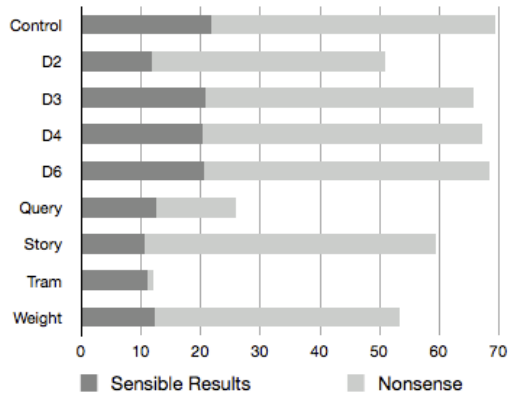

Figure 5: Unique sensible versus nonsensical results.

alternate query was completely blank. As figures 4 and 5 show, this alternate query generated fewer unique results, but had much better ratios of s/ns in both the unique results (.947) and the total results (3.103). Given our TRAMs and story library, this alternate query is apparently more restrictive, but also makes it easier for the system to find sensible results. As might be expected, this increased story quality comes with a decrease in story variance: figure 6 shows that the expected differences between two results using the alternate query has fallen to about 3.5 (which is approximately half of the 7 differences expected from the baseline). The query test shows that our example query should not be taken as a good estimate of the average query: there is a lot of variation between queries, and we have no reason to expect that our default query is representative. This test also exhibits the fundamental trade-off of our system: more constrained queries increase the s/ns ratio, but come at the expense of less variation. To overcome this trade-off, the system would have to either cull nonsensical results, or simply generate variety that does not include nonsense.

After testing varying the query, we next tested the effect of library size. We constructed a smaller story library by randomly removing eight stories from our default library. We then re-ran the original query (the results are listed under Story in our graphs). This smaller story set significantly reduced the number of unique stories generated, while at the same time depressing the s/ns ratio. The expected number of differences remained high, but removing stories clearly degrades the system: MINSTREL has trouble finding parsimonious story fragments during TRAM searches, and as a result it more often resorts to nonsensical matches. Interestingly, though, the decrease in the number of unique results was not proportional to the decrease in the number of stories (although the decrease in the number of sensible unique results was). This implies that nonsensical stories are easier to generate than sensible ones.

Our next test was the most promising: we took the two TRAMs that create the most liberal changes (SimilarOutcomes and GeneralizeConstraint) and removed them. The results clearly show how important TRAMs are: the sense to nonsense ratio was drastically increased for both total and unique results. At the same time, they show even more clearly that nonsense is the usual price for variation: although the majority of results were now sensible, the variation within these results was reduced: the TRAM case has an expected differences value of close to 3, compared to the original 7. Effectively, even though these liberal TRAMs often create nonsense, they are also key in creating enough variation to generate creative results.

For our final trial, we implemented an alternate search method that we hoped would provide a compromise between the chaos of the more liberal TRAMs and the boring results generated without them. Rather than use random TRAM selection during search, we used weighted random selection, and biased the selection towards less-liberal TRAMs. We gave the two liberal TRAMs that had been removed in the TRAM case weights of 2 and 3, and most of the TRAMs got a weight of 5. Two of the more specific TRAMs got weights of 8 and 10. Given these weights, the expected variation was maintained, but the s/ns ratios decreased for both total and unique results. The number of unique results decreased as well. Figure 7 shows that among all results, results that were found after only a single TRAM application significantly increased, at the expense of results that used more than 4 TRAMs. Interestingly, the distribution of unique results among depths (seen in figure 8) did not fundamentally change. Essentially, our weighting scheme did not help find new results, but instead biased the total results towards shallow unique results, some of which were nonsensical. This result demonstrates that even minor changes to the search method can negatively impact the results (as opposed to simply favoring either variety or quality). The fact that TRAM weights can significantly impact the result set also suggests that a principled approach to selecting TRAM weights could potentially enhance the system.

Conclusions

To evaluate the creativity of MINSTREL's TRAM system, we adopted formal measures of variety and quality to systematically investigate the effect of MINSTREL parameters and design choices on the creativity of the system. In addition to indicating measurable creativity we were pleased to notice that a number of the story fragments showed interesting results (for example, one of our stories involved a group of possessed townsfolk: the princess wants to injure them because they are possessed, but ends up killing them).

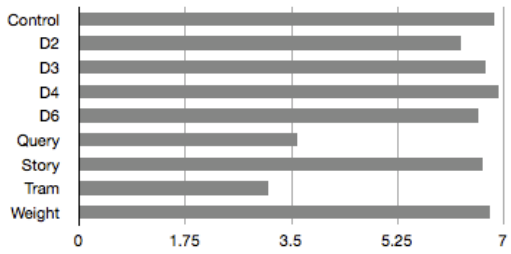

Figure 6: The expected number of fields that will differ between two results.

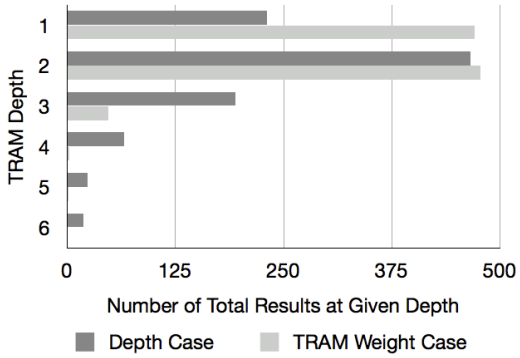

Figure 7: TRAM depths for the D6 and Weight trials.

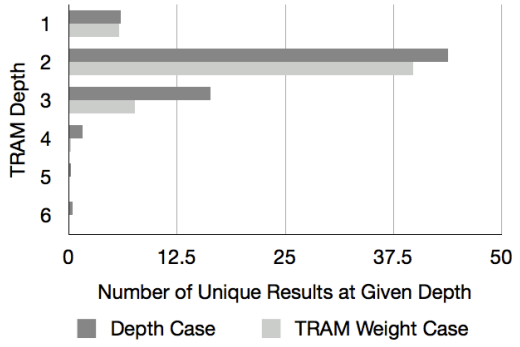

Figure 8: Unique depths for the D6 and Weight trials.

We feel that our main finding is that there is a demonstrable trade-off between the variety and the quality of the results: conditions that increase the quality of the results come with corresponding decreases in the variety thereof. No single configuration among our tests emerged as superior, but given a preference for either variance or sensibility, we have enough data to recommend a set of parameters. By measuring both variance and quality, we were able to effectively track the creative properties of the system, and observe both expected and unexpected changes under various conditions. Although our measurement of creativity is not direct, our data-driven approach has allowed us to make very specific statements about the way that the system operates, and to experiment and discover nuances of the system that would not be uncovered with a more general creativity assessment.

Future Work

The next step in the evolution of MINSTREL REMIXED is to implement Turner's author-level planning system. Creativity in MINSTREL is not restricted to the TRAM system alone: the way that author-level plans use the TRAM system and the interplay between various author-level plans gives rise to sensible creative results (for example, there are author-level plans that check the consistency of the story, which can be helpful when the TRAM system comes up with odd results). Once author-level planning is implemented, a more holistic study of Minstrel's creativity could be produced. To do this, more modern measures of creativity such as cognitively inspired methods (Riedl and Young 2006) and non-automated evaluative frameworks (Ritchie 2007) should be investigated to determine what measures would be applicable to the full system output.

We may also decide to revisit our TRAM weighting system. Although the weights that we chose for this experiment resulted in poor performance, armed with metrics for variance and quality, we could optimize the TRAM weights. This process would provide further information about how each TRAM contributes to variation and to sensibility.

Acknowledgements

This work was supported in part by the National Science Foundation, grants IIS-0747522 and IIS-1048385. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Boden, M. 2004. *The Creative Mind*. London: Routledge, 2nd edition.

Fairclough, C., and Cunningham, P. 2003. A multiplayer case based story engine. Technical report, Trinity College Dublin, Department of Computer Science, Dublin.

Gervas, P.; Diazagudo, B.; Peinado, F.; and Hervas, R. 2005. Story plot generation based on CBR. *Knowledge-Based Systems* 18(4-5):235–242.

Musen, M. a.; Gennari, J. H.; and Wong, W. W. 1995. A rational reconstruction of INTERNIST-I using PROTEGE-II. In *Proc. Annual Symp. on Computer App. in Medical Care.*, 289–93.

Pease, A., and Winterstein, D. 2001. Evaluating machine creativity. In *Proc. ICCBR. Workshop on Creative Systems*, 129–137.

Peinado, F., and Gervas, P. 2006. Minstrel Reloaded : From the Magic of Lisp to the Formal Semantics of OWL. *Lecture Notes in Computer Science* 4326(2006):93–97.

Riedl, M., and Young, M. 2006. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Generation Computing* 24(3):303–323.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99. 10.1007/s11023-007-9066-2.

Smith, G.; Whitehead, J.; Mateas, M.; Treanor, M.; March, J.; and Cha, M. 2011. Launchpad: A Rhythm-Based Level Generator for 2D Platformers. *To appear in IEEE TCIAIG* 0(0).

Tearse, B.; Mateas, M.; and Wardrip-Fruin, N. 2010. MINSTREL Remixed: a rational reconstruction. In *Proc. of the INT III Workshop*. Monterey, CA: ACM.

Turner, S. R. 1994. *The Creative Process: A Computer Model of Storytelling And Creativity*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.