Creative Help: A Story Writing Assistant

Melissa Roemmele and Andrew S. Gordon

Institute for Creative Technologies
University of Southern California
Los Angeles, CA, USA
roemmele@ict.usc.edu, gordon@ict.usc.edu

Abstract. We present Creative Help, an application that helps writers by generating suggestions for the next sentence in a story as it being written. Users can modify or delete suggestions according to their own vision of the unfolding narrative. The application tracks users' changes to suggestions in order to measure their perceived helpfulness to the story, with fewer edits indicating more helpful suggestions. We demonstrate how the edit distance between a suggestion and its resulting modification can be used to comparatively evaluate different models for generating suggestions. We describe a generation model that uses case-based reasoning to find relevant suggestions from a large corpus of stories. The application shows that this model generates suggestions that are more helpful than randomly selected suggestions at a level of marginal statistical significance. By giving users control over the generated content, Creative Help provides a new opportunity in open-domain interactive storytelling.

Keywords: open-domain interactive narrative · writing aids · natural language generation

1 Introduction

The field of artificial intelligence has long conceived of using computers to write stories. The first known automated story generation system, *Novel Writer*, was developed in 1973 [7], followed by a steady line of work up to the present [e.g. 8, 13, 15]. These systems act as writing agents that generate narratives from hand-authored models of the characters, settings, and actions comprising the story-world domain. Similarly, there are works of fiction generated through user interaction, with these first of these systems, *Adventure*, emerging in 1975 [2]. Despite the apparent user-driven nature of interactive fiction systems, they are similar to the AI systems in their autonomy. They expect users' writing to adhere to a highly constrained syntax and vocabulary in order to continue the emerging narrative. They also similarly rely on hand-authored domain models that push users towards one of a limited number of predefined experiences. The current challenge for narrative AI is to provide interactivity that gives human authors control over writing, enabling them to write the stories they want to tell.

One of the barriers to advancing true interactivity is automatically understanding the author's intended meaning as the story is being written. Advances in the field of natural language processing are starting to break down this barrier. Many emerging language technologies can be seen as assistive tools that recognize users' intent and help them achieve it. Automated spelling and grammar correction, as commonplace as they seem, are examples of simple writing aids. We envision a new sort of writing aid, one that performs "narrative autocompletion": it analyzes a story as it is being written and then makes a suggestion for how to continue the story.

Recent work in open-domain interactive storytelling now supports this vision. The impracticality of authoring models for every possible story domain has motivated the effort to learn such models from data [1, 11]. Swanson and Gordon [19] describe a case-based reasoning approach for generalizing knowledge from existing stories to new ones. They present this approach as Say Anything, a platform where the user and automated agent take turns writing sentences in a story. The success of this system in producing readable stories established a new path towards users having authorial control in interactive narrative experiences.

In this paper, we present Creative Help, a system that builds on Swanson and Gordon's story generation approach [19], but is unique in its role as an assistive tool that places the writer in control of the resulting narrative. Writers use Creative Help to generate ideas for what happens next in their story. In contrast to Say Anything, users both choose when to request a suggestion and what to do with the requested suggestion, enabling them to maintain the primary influence over the outcome of the story.

Previous systems have struggled to provide an objective evaluation of generated content, relying on questionnaire-based tasks where people read and rate the stories. In contrast to this, Creative Help has an inherent capacity to evaluate its own output by tracking users' revision of the generated suggestions. The application measures the similarity between the suggestions and users' modifications to them, providing a quantified score of users' interest in the suggestions. We use this functionality to compare different models for retrieving suggestions, which in turn enables us to draw conclusions about each model's relative helpfulness to writers. From these results we discuss how to move towards an automated story writing assistant that maximizes the potential of authors' storytelling creativity.

2 Creative Help

Creative Help is a web-based application for writing stories. Users request "help" from the application to automatically author a new sentence. The user interface (Figure 1) is extremely simple: users see a text box where they can start typing a story. When the user wants to receive a suggestion for the next sentence in the story, she types "\help\". This initiates a Javascript function that sends a request to a CherryPy back-end server to generate a sentence. The server has a generation model that accesses a large corpus of stories in order to find a suggestion to fulfill the help request. The suggested sentence appears in the text

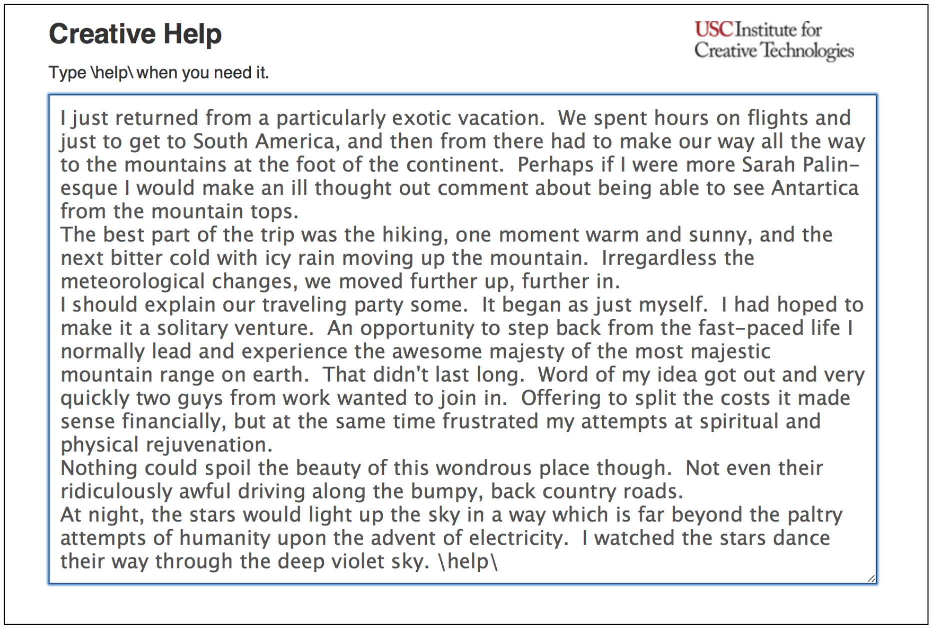

Fig. 1. Creative Help

in place of the "\help\" string. The user can modify this suggestion like any other text that already appears in the story. She can then continue writing, making additional requests for suggestions whenever she chooses. The application tracks the user's changes to suggested sentences in order to evaluate their quality as contributions to the story. We explain the details of this pipeline below.

2.1 Generating Suggestions

The mechanism by which Creative Help generates suggestions is similar to the generation component of SayAnything, a system that takes turns with a human user in writing sentences in a story [19]. In SayAnything, the approach was to search within a large corpus of stories to find the sentence that is most similar to the sentence the user wrote in his most recent turn. Once this sentence is located, the system retrieves the sentence that immediately follows it in the corresponding story. This is the sentence that is contributed as the system's turn. Analogously, in Creative Help, the sentence that appears directly before the help request is the sentence for which the system finds a most similar match. This technique follows the philosophy of case-based reasoning, which draws inferences about a new instance by comparing it to one that has been observed before. Here, the system infers that the generated sentence is a good continuation of the story because it appears in a story similar to the one that has been written so far.

In order to compute similarity between sentences, we use the software package Apache Lucene [5] to index all words in the corpus. An index stores the number of times a word occurs in a sentence (called *term frequency*) as well as all the sentences in which a word occurs (called *document frequency*). This information enables Lucene to encode a sentence as a vector of words, with each word represented by its term frequency-inverse document frequency (tf-idf) weight. The tf-idf weight scheme assigns higher weight to words that occur more frequently in the sentence relative to their frequency overall in the corpus. This representation, called the Vector Space Model, enables efficient computation of similarity between sentences in terms of their vector similarity. Given the sentence the user wrote before making the help request, Lucene scores its similarity to all other sentences in the corpus using a formula based on cosine similarity. The sentence with the highest similarity score is selected as the most similar match. Along with the text of a sentence, we store the ID of the sentence that follows it. With this information, the system can easily retrieve the sentence that occurs after the most similar match and return it to the Creative Help user as the suggested continuation of their story.

2.2 Data

The suggestions for Creative Help are generated from a corpus of approximately twenty million English-language stories. These stories were identified using a story classification tool developed by Gordon and Swanson [4], which was used to specifically extract stories from a large set of public weblog posts authored between January 2010 and August 2014. We segmented each of these stories into sentences using the Stanford CoreNLP sentence tokenizer [12], which detects sentence boundaries at sentence-ending characters (".", "!", or "?") that are not contained in a token such as an abbreviation. Using this tool, the 20,337,098 stories in the corpus were segmented into a total of 681,921,109 sentences.

2.3 Modifying Suggestions

As soon as the suggested sentence appears to the user, the application starts tracking any edits the user makes to the sentence. Specifically, the JavaScript component of the interface listens for any keystroke events occurring in the text area where the suggestion appears. If the majority of the text characters are removed from the suggestion (such that less than ten characters remain), the application considers the suggestion to have been deleted. Otherwise, the application continues to track edits to the suggestion. If a suggestion has remained unchanged for at least one minute, the tracking to that sentence "expires" and it is assumed that the user has made his final modifications to the sentence. As soon as a suggestion is given a deleted or expired status, it is logged to a SQLite database along with its original form before it was modified. For deleted suggestions, the modified sentence is just an empty string. In some cases, the suggestion might be "lost" in that the application is no longer able to find the tracked location where the suggestion first appeared. This might happen if the user suddenly cuts and pastes over all the text in the story, for instance. If this happens, the suggestion is assigned a lost status and logged accordingly in the

"Last night I had a crazy dream."
"A strange thing happened on my way home yesterday."
"I had the most awkward dinner of my life last night."
"I received a surprising phone call yesterday."
"Last week some old friends came in town for a visit."
"Last year I took a cross-country road trip."
"Last weekend I went to a party at a friend's house."
"I just returned from a particularly exotic vacation."
"I was sitting in my desk at work when I saw the news."
"I got in trouble a lot when I was younger."
"As a kid, I once made an unusual discovery."
"The scar on my leg has an interesting story."
"My first day of high school was unforgettable."
"Last night's performance was spectacular."
"On April Fools' Day, my co-workers played a prank on me."
"My friends planned a surprise party for my birthday."
"I rarely get angry, but yesterday was one of those days."
"I bumped into my ex a few weeks ago."
"I recently decided to make a big change."
"This morning I noticed a stranger staring at me."

Table 1. List of suggested first sentences, not included in any generation model

database. As the next section explains, this tracking data directly shows the comparison between a suggestion and its modification, revealing how helpful the user judged the suggestion to be.

2.4 Story Initialization

When a user first starts writing her story, there is no previous sentence from which to generate a suggestion. For help beginning a story, we wrote a list of twenty introductory sentences that could in turn generate a reasonable continuation. This list, shown in Table 1, illustrates the type of stories most highly represented in our corpus: personal experiences narrated in the first person. If the user types "\help\" before anything else, we randomly pick a sentence from our list as the suggested first sentence in her story. Users can edit these sentences like any other part of the story, but they are not evaluated in association with any generation model; we simply skip over these suggestions in our analysis.

3 Experiment

In allowing users to modify suggestions, Creative Help contains a built-in mechanism for evaluating the suggestion generation system. We assume that if users consider suggestions to be good contributions to the story, they will edit them less frequently than suggestions they don't consider helpful to the story. Thus, variations in the quality of the model for generating suggestions should be revealed by differences in the rate of users' modifications. We designed an experiment to further explore this idea. In our experiment, users wrote stories with the Creative Help application and each time a user requested help, we randomly varied the model for retrieving the suggestion. We selected four models that we expected to differ based on the quality of suggestions they provided.

3.1 Models

We evaluated four models, which we refer to as the *full* model, the *reduced* model, the *diegetic* model, and the *random* model. The first three models all use the scheme described in the previous section: they consider the sentence the user wrote directly before typing "\help\" and search for the sentence in the corpus with the highest vector-based similarity. What distinguishes these models is the corpus from which they retrieve sentences. In the full model, a similarity match is found among the full corpus of approximately 20 million stories (680 million sentences). In the reduced model, sentences are retrieved from a reduced subset of the corpus containing one million stories, which was the size of the corpus used in Say Anything [19]. The purpose of comparing these two models is to observe if the amount of story data influences the quality of suggestions. We hypothesize that increasing the number of stories will yield more precise similarity matches, making it more likely that the generated suggestion fits the story.

In our third model, the diegetic model, we explore whether certain sentences of stories are better for generating suggestions than other sentences of the same story. In particular, each part of a story can refer to one of two narrative levels: the *diegetic* level on which the events in the story take place, or the *extradiegetic* level on which the narration of the story takes place. For example, consider the first two sentences in the following story. In the first, the writer is describing her experience as a character within the story at the time it occurred. In the second sentence, the writer is addressing the reader at the present moment (the time of writing).

> I walked to yesterday's party with my friend around 9pm. *(diegetic)*
> Normally I try to stay away from big parties. *(extradiegetic)*

Rahimtoroghi et al. [16] proposed that because they focus on story-world events over evaluative commentary, diegetic sentences are more informative than extradiegetic sentences in continuing a story. We explored this possibility without having to manually annotate every sentence in the corpus. We used Sagae et al.'s tool [17], which automatically predicts a label of "diegetic" or "extradiegetic" for each sentence in a story based on linguistic features. Our diegetic model evaluates the hypothesis that specifically targeting the diegetic sentences in stories will yield more helpful suggestions. This model finds a similarity match among only the sentences labeled as diegetic in the full corpus of 20 million stories; the returned suggestion must also be labeled diegetic.

Finally, we compare these first three models to a fourth model, the random model, which just randomly selects a sentence from the corpus as the suggestion.

Since the random model has no knowledge of the user's story, we expect users to judge these suggestions as poor relative to the models that take the most recent sentence of the user's story into account.

Based on the idea that better suggestions will receive fewer edits, we predict differences between each of these four models in terms of how much users modify the suggestions they generate. Obviously, we expect the random model to perform the worst. Because it considers the user's story, we expect the reduced model to outperform the random model, but perform worse than the full or diegetic model because it has fewer stories from which to generate suggestions. Finally, we predict that the full model will perform better than random and reduced models, but that the best suggestions will come from the diegetic model. Our reasoning is that compared to the full model, the diegetic model specifically targets the sentences in stories that are most relevant for generating suggestions. To summarize, we hypothesize the following ordering in the models' quality of suggestions: random $<$ reduced $<$ full $<$ diegetic.

3.2 Task

We recruited 24 people to use Creative Help as part of our experiment. Users were all employees of our research organization who responded to an email inviting them to "try out a computer-assisted story writing tool". Users came to our lab to participate so that we could instruct them about the task in person and respond to any potential questions or issues they had while using the application. We deliberately gave participants only a few instructions: we told them to write a story about anything they wanted to write about, and to type "\help\" when they wanted to receive a suggestion for the next sentence in the story. We made it clear that they could choose to modify, add to, or delete the suggestion however they wished. We asked them to spend 20 minutes writing, but they were free to continue using the tool for longer. Each time a user requested help, the application randomly selected one of the four generation models, so users were equally likely to receive suggestions from any one of the models. Users were not aware of the method by which the suggestions were generated, other than by observing the suggestions themselves. At the end of the task users had the opportunity to provide feedback and ask questions about the purpose of the tool.

3.3 Evaluation

As discussed, by recording suggestions before and after they have been modified by users, Creative Help affords a means of evaluating the quality of suggestions. We presume an inverse relation between a user's edits to a suggestion and the quality attributed to it. In order to objectively determine if there were any differences between the models in terms of this relation, we needed a method for quantifying the difference between the original and modified form of a suggestion. In fact, there is a popular method in computer science for measuring the similarity between two strings: edit distance. Edit distance encodes the similarity between two strings in terms of the number of operations required to transform

Model	Edit distance	Deletion rate
random	0.760	0.628
reduced	0.729	0.573
full	**0.672**	**0.516**
diegetic	0.718	0.556

Table 2. Mean normalized edit distance and deletion rate of suggestions by model

one string into another, where an operation is an addition, deletion, or substitution of a single character. The more similar two strings are, the lower their edit distance will be; an edit distance of 0 indicates the strings are identical, while an edit distance equal to the length of the longer string indicates the strings share no common characters. There are a few algorithms for computing edit distance, but the traditional one is Levenshtein edit distance [10], which is what we used for computing the similarity between each original and modified suggestion. Because the length of a suggestion affects the edit distance score, we divided the Levenshtein distance by whichever sentence was longer, the original suggestion or its modification. This results in a normalized edit distance score between 0 and 1 [6], where 0 indicates that no edits were made to the suggestion and 1 indicates that the suggestion was entirely deleted or replaced. By computing the edit distance between a suggestion before and after the user has modified it, we can compute the mean edit distance of each model's suggestions as a way of scoring the quality of the model. A lower mean edit distance for a model means that users made fewer edits to that model's suggestions, presumably because they found those suggestions more helpful for continuing their story.

4 Results

There were a total of 378 suggestions requested across all 24 users, an average of about 16 suggestions per user, with each user receiving on average 4 suggestions from each model. The mean normalized Levenshtein distances for the suggestions generated by each model appear in Table 2. We also computed the proportion of suggestions that were deleted for each model, which shows how frequently users deleted suggestions. Even for the full model, where suggestions were edited the least, users deleted half (51.6%) of the suggestions. The deletion rates and edit distances demonstrate the same pattern across the models. As we predicted, the random model has the highest mean normalized edit distance, meaning that its suggestions were edited more often than those of the other models. We used the compute-intensive randomized test with stratified shuffling [14] in order to evaluate the hypothesized differences between the models. The p-values computed by this method represent the probability of observing model differences at least as large as the ones shown in Table 2 if there were no actual differences. We found a marginally significant difference between the full model and the random model in terms of both normalized edit distance and deletion rate ($p=.072$ and $p=.090$, respectively), with the suggestions generated by the

(1) Once upon a time... In a galaxy far, far away... a donut shop on the planet Xantofar was experiencing a crisis. *Amber, the sweet but apathetic shop worker, had run out of flour to make the shop's delicious donuts.* **Suggestion (diegetic):** "After that we headed a few miles down the road to Davis Bayou National Park." **Modification:** "She tried going to a grocery store a few miles down the road near Davis Bayou National Park, but the manager said that due to a rare flour shortage the whole county was completely out."

(2) "I recently decided to make a big change. I decided to become a super hero. It all began when I spray painted a bunch of boxes black and put squares of yellow on them to look like buildings and then I used my boys superhero toys to help decorate. *I dressed up like Godzilla and started rampaging through this tiny, helpless box town."* **Suggestion (diegetic):** "Vanessa made for great stories, but I related to-and sort of sided with-my mom." **Modification:** *(deleted)*

(3) "Last year I took a cross-country road trip. Bought a 15 day pass and (after visiting the Grand Canyon), caught the Southwest Chief in Flagstaff. *And then I met a prostitute."* **Suggestion (full):** "He made me feel wonderful at first." **Modification:** *(no changes)*

(4) "This weekend my girlfriend and I took the train from Los Angeles to San Francisco. Our journey started out wonderfully, and we found the train ride to be particularly relaxing and picturesque. *Unfortunately, we hit a snag in Oxnard: the train broke down in the station."* **Suggestion (full):** "Madi, can we go see the Palarie dogs?" **Modification:** "Mari, can we go see the prairie dogs?"

(5) *"Once upon a time there was a cabin in an island"* **Suggestion (random):** "Please keep reading my blog!" **Modification:** *(deleted)*

(6) "Once upon a time there was a young couple who met in a dream. *Each thought the other was a dream mate and not real, until"* **Suggestion (random):** "The location was also pretty good and we could easily walk to the main attractions in Foshan without having to use our tourist guide." **Modification:** "they landed in a dream location that was so pretty ugly that they wondered if they were actually dreaming or had been transported to another reality."

(7) "I received a surprising phone call yesterday. *It was my parole officer.*" **Suggestion (reduced):** 'We talked about work, how my anger was soothed, seduced by the dog." **Modification:** *(no changes)*

(8) "A girl becomes entranced by an inkling of an idea. *She starts to think about how we are all connected and how each interaction and passing we have with one another impacts all those around us."* **Suggestion (reduced):** "It was a challenge like no other." **Modification:** *(deleted)*

Fig. 2. Examples of Creative Help suggestions

full model being modified less than the random suggestions. We take this as evidence that suggestions incorporating the content of the user's story are more helpful than generic suggestions. This may seem obvious, but it validates the use of Creative Help as an evaluation platform. By tracking users' changes to the output of each generation model and quantifying those changes with a simple similarity metric (edit distance), our application revealed an objective difference in the quality of the models.

Our results did not show a significant impact of the size of the corpus on the generation model, given that the model with access to 20 million stories did not provide better suggestions than the model using only 1 million stories. However, given that the full model did perform the best, it's possible that such a difference would emerge with a larger set of users. Moreover, the hypothesis that diegetic sentences would be more helpful did not hold up. One interpretation is that filtering the corpus of extradiegetic sentences has the same effect as reducing the corpus size, merely limiting the number of examples from which to infer a fitting continuation of the user's story. It's also likely that users issued help requests upon contributing extradiegetic sentences, so the model was mismatching extradiegetic sentences with diegetic ones. An improved model would take the narrative level of the user's contribution into account and search for a similarity match of the same narrative level.

Figure 2 shows examples of suggestions originally generated by each model and their resulting form after being modified by the user. Each example shows the story written so far when the user made the help request, as well as the returned suggestion and how the user ultimately modified it. The sentence that preceded the help request (and thus generated the suggestion) is italicized. Even with the high number of deletions (e.g. examples 2, 5, and 8), some suggestions were retained without any editing (examples 3 and 7). Among the modified ones, users employed different strategies for rewriting these sentences. In some cases, they simply replaced the names of entities (example 4). In other cases, the modification had a related meaning to the original, with some extended or altered details (example 1). Other suggestions were completely transformed, expressing a new meaning even if retaining a few phrases from the original (example 6).

5 Discussion and Future Work

In this work, we defined a new paradigm of "narrative auto-completion", which assists users in writing the next sentence in a story. This functionality affords users more control over an emerging story than previous interactive narrative systems where system-generated content is entirely fixed. Moreover, by enabling users to modify generated content, Creative Help natively evaluates the quality of that content. We used this functionality to compare different generation models and showed that by modeling the domain of the emerging story, we can improve suggestions offered to users. We generated suggestions using case-based reasoning to find similarities between the user's ongoing story and stories in a massive corpus, but there are many alternative designs for generation models. Our goal is not to argue that the full model defined in this work is optimal, but to show that alternative models can be comparatively evaluated using Creative Help.

We compared models in terms of how much users edited their suggestions to incorporate them into the narrative, with the idea that more helpful suggestions receive fewer edits. The next step is to examine what makes a suggestion more or less helpful to a story. In the case where the suggestion was deleted, there are many possible reasons this occurred. The suggestion could have been entirely incoherent with the story (e.g. example 5 in Figure 2), or the user might have just disliked the suggestion despite it making sense within the story. On the other hand, we can assume that if a suggestion or some part of it was retained, the user found that part to be a helpful contribution to the story. We presented some characteristic examples in Figure 2 of how users adapted these suggestions. As future work we plan to analyze what features promote a suggestion's adaptability to the story. In SayAnything [19], maximizing the local coherence between the generated sentence and the user's most recent contribution improved the acceptability of the system's contributions. We expect this to be true in Creative Help as well, but the modification data provides a new opportunity for insight.

This insight will help us refine our target for improving the full generation model used here, which will hopefully increase the rate at which suggestions are retained. One immediate plan is to allow users to request help for completing a sentence that is already partially written. In looking at the data, we noticed that many users expected this functionality, but the models were instead designed to return a new sentence. Another clear need to be addressed is incorporating the context of the full story into the generation model. The current model uses only the most recent sentence to generate the suggestion, which leads to weak global coherence between the generated sentence and the story as a whole. Swanson and Gordon [18] discuss this challenge, which lies in appropriately weighting the contextual information so that it does not compromise the local coherence of the suggestion. Looking farther forward, we want to take advantage of recent progress made in natural language processing on tasks like semantic role labeling [3], distributional word representations [20], and coreference resolution [9]. We expect that by annotating our story corpus with this linguistic information, we can represent stories at a more abstract level than the shallow word-based representation used in our current models. In doing so, we hope this will maximize the completeness and accuracy of the generation system's narrative knowledge. Creative Help can then use this knowledge to offer writers new and perhaps previously unimagined possibilities for storytelling.

6 Acknowledgments

The projects or efforts depicted were or are sponsored by the U. S. Army. The content or information presented does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

1. Chambers, N., Jurafsky, D.: Unsupervised learning of narrative event chains. In: 46th Annual Meeting of the Association of Computational Linguistics. pp. 789–797 (2008)
2. Crowther, W., Woods, D., Black, K.: Colossal cave adventure (1976)
3. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Computational Linguistics 28(3), 245–288 (2002)
4. Gordon, A.S., Swanson, R.: Identifying Personal Stories in Millions of Weblog Entries. In: 3rd International Conference on Weblogs and Social Media, Data Challenge Workshop. pp. 16–23 (2009)
5. Hatcher, E., Gospodnetic, O., McCandless, M.: Lucene in action. Manning Publications (2004)
6. Heeringa, W.J.: Measuring dialect pronunciation differences using Levenshtein distance. Ph.D. thesis, Rijksuniversiteit Groningen (2004)
7. Klein, S., Aeschlimann, J., Balsiger, D.: Automatic novel writing: A status report. Wisconsin University (1973)
8. Lebowitz, M.: Story-telling as planning and learning. Poetics 14(6), 483–502 (1985)
9. Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., Jurafsky, D.: Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules. Computational Linguistics 39(4), 885–916 (2013)
10. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady 10, 707–710 (1966)
11. Li, B., Lee-Urban, S., Johnston, G., Riedl, M.: Story Generation with Crowdsourced Plot Graphs. In: 27th AAAI Conference on Artificial Intelligence (2013)
12. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. In: 52nd Annual Meeting of the Association for Computational Linguistics. pp. 55–60 (2014)
13. Meehan, J.R.: TALE-SPIN, An Interactive Program that Writes Stories. In: 5th International Joint Conference on Artificial Intelligence. pp. 91–98 (1977)
14. Noreen, E.W.: Computer intensive methods for hypothesis testing: An introduction (1989)
15. Pérez y Pérez, R., Sharples, M.: MEXICA: A computer model of a cognitive account of creative writing. Journal of Experimental & Theoretical Artificial Intelligence 13(2), 119–139 (2001)
16. Rahimtoroghi, E., Corcoran, T., Swanson, R., Walker, M.A., Sagae, K., Gordon, A.S.: Minimal Narrative Annotation Schemes and Their Applications. In: 7th Intelligent Narrative Technologies Workshop (2014)
17. Sagae, K., Gordon, A.S., Dehghani, M., Metke, M., Kim, J.S., Gimbel, S.I., Tipper, C., Kaplan, J., Immordino-Yang, M.H.: A Data-Driven Approach for Classification of Subjectivity in Personal Narratives. In: 2013 Workshop on Computational Models of Narrative. pp. 198–213 (2013)
18. Swanson, R., Gordon, A.S.: A Comparison of Retrieval Models for Open Domain Story Generation. In: AAAI Spring Symposium on Intelligent Narrative Technologies II (2009)
19. Swanson, R., Gordon, A.S.: Say Anything: Using Textual Case-Based Reasoning to Enable Open-Domain Interactive Storytelling. ACM Transactions on Interactive Intelligent Systems 2(3), 1–35 (2012)
20. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: 48th Annual Meeting of the Association for Computational Linguistics. pp. 384–394 (2010)