Linguistic Features of Helpfulness in Automated Support for Creative Writing

Melissa Roemmele and Andrew S. Gordon
Institute for Creative Technologies, University of Southern California
roemmele@ict.usc.edu, gordon@ict.usc.edu

Abstract

We examine an emerging NLP application that supports creative writing by automatically suggesting continuing sentences in a story. The application tracks users' modifications to generated sentences, which can be used to quantify their "helpfulness" in advancing the story. We explore the task of predicting helpfulness based on automatically detected linguistic features of the suggestions. We illustrate this analysis on a set of user interactions with the application using an initial selection of features relevant to story generation.

1 Introduction

At the intersection between natural language generation, computational creativity, and human-computer interaction research is the vision of tools that directly collaborate with people in authoring creative content. With recent work on automatically generating creative language (Ghazvininejad et al., 2017; Stock and Strapparava, 2005; Veale and Hao, 2007, e.g.), this vision has started to come to fruition. One such application focuses on providing automated support to human authors for story writing. In particular, Roemmele and Gordon (2015), Khalifa et al. (2017), Manjavacas et al. (2017), and Clark et al. (2018) have developed systems that automatically generate suggestions for new sentences to continue an ongoing story.

As with other interactive language generation tasks, there is no obvious approach to evaluating these systems. The number of acceptable continuations that can be generated for a given story is open-ended, so measures that strictly rely on similarity to a constrained set of gold standard sentences, e.g. BLEU score (Papineni et al., 2002), are not ideal. Moreover, the focus of evaluation in interactive applications should be on users' judgments of the quality of the interaction. While it is straightforward to ask users to rate generated content (McIntyre and Lapata, 2009; Pérez y Pérez and Sharples, 2001; Swanson and Gordon, 2012), self-reported ratings for global dimensions of quality (e.g. "on a scale of 1-5, how coherent is this sentence in this story?") do not necessarily provide insight into the specific characteristics that influenced these judgments, which users might not even be explicitly aware of. It is more useful to examine users' judgment on an implicit level: for example, by allowing them to adapt generated sequences. This is related to rewriting tasks in other domains like grammatical error correction (Sakaguchi et al., 2016), where annotators edit sentences to improve their perceived quality. This enables the features of the modified sequence to be compared to those of the original.

In this work, we analyze a set of user interactions with the application Creative Help (Roemmele and Gordon, 2015), where users make 'help' requests to automatically suggest new sentences in a story, which they can then freely modify. We take advantage of Creative Help's functionality that tracks authors' edits to generated sentences, resulting in an alignment between each original suggestion and its modified form. Previous work on this application compared different generation models according to the similarity between suggestions and corresponding modifications, based on the idea that more helpful suggestions will receive fewer edits. Here, we focus on quantifying suggestions according to a set of linguistic features shown by existing research to be relevant to story generation. We examine whether these features can be used to predict how much authors modify the suggestions. We propose that this type of analysis is useful for identifying the aspects of generated content authors implicitly find most helpful for writing. It can inform the evaluation of future creativity support systems in terms of how

well they maximize features associated with helpfulness.

2 Application

The Creative Help interface consists simply of a text box where users can write a story. Authors are instructed that they can type *help*\ at any point while writing in order to generate a suggestion for a new sentence in the story, and that they can freely modify this suggestion like any other text that already appears in the story. As soon as the suggested sentence appears to the author, the application starts tracking any edits the author makes to it. Once one minute has elapsed since the author last edited the sentence, the application logs the modified sentence alongside its original version. See Roemmele and Gordon (2015) for further details about this tracking and logging functionality. The result of authors' interactions with the application is a dataset aligning generated suggestions to their corresponding modifications along with the story context that precedes the help request.

The current generation model integrated into Creative Help is a Recurrent Neural Network Language Model (RNN LM) with Gated Recurrent Units (GRUs) that generates sentences through iterative random sampling of its probability distribution, as described in Roemmele and Gordon (2018). The motivation for this baseline model is that by training it on a corpus of fiction stories, it produces sequences that are likely to appear in these stories, but the unpredictability associated with random sampling yields novel word combinations that may be appealing from the standpoint of creativity (Boden, 2004; Dartnall, 2013; Liapis et al., 2016). The RNN LM was trained on a subset of the BookCorpus[1] (Kiros et al., 2015), which contains freely available fiction books uploaded by authors to smashwords.com. The subset included 8032 books from a variety of genres, which were split into 155,400 chapters (a little over half a billion words). To prepare the dataset for training, the stories were tokenized into lowercased words. All punctuation was treated in the same way as words. A vocabulary of all words occurring at least 25 times in the text was established, which resulted in 64,986 unique words being included in the model. All other words were mapped to a generic <UNKNOWN> token that was restricted from being generated. Proper names were handled uniquely by replacing them with a token indicating their entity type and a unique numerical identifier for that entity (e.g. <PERSON1>). During generation, a list of all entities mentioned prior to the help request was maintained. When the model generated one of these abstract entity tokens, it was replaced with an entity of the corresponding type and numerical index in the story. If no such entity type was found in the story, an entity was randomly sampled from a list of entities found in the training data.

The RNN[2] was set up with a 300-dimension word embedding layer and two 500-dimension GRU layers. It was trained for one single iteration through all chapters, which were observed in batches of 125. The Adam algorithm (Kingma and Ba, 2015) was used for optimization. To generate a sentence when a help request was made, the model observed all text prior to the help request (the context) to compute a probability distribution for the next word. A word was sampled from this distribution, appended to the story, and this process was repeated to generate 35 words. All words after the first detected sentence boundary[3] were then filtered (in some cases, no sentence boundary was detected so all 35 words were included in the returned sentence). Finally, the suggestion was 'detokenized' using some heuristics for punctuation formatting, capitalization, and merging contractions before being presented to the author.

3 Experiment and Analyses

We recruited people via social media, email, and Amazon Mechanical Turk to interact with Creative Help[4] for at least fifteen minutes. Participants were asked to write a story of their choice. They were told the objective of the task was to experiment with asking for \help\ but they were not required to make a certain number of help requests. They could choose to edit, add to, or delete a suggestion just like any other text in their story, without any requirement to change the suggestion at all. Ultimately, 139 users participated in the task, resulting in suggestion-modification pairs for 940 help requests, which includes pairs where the suggestion and modification are equivalent because no edits were made.

Given this dataset of pairs, we first quantified

[1] yknzhu.wixsite.com/mbweb

[2] Code at: github.com/roemmele

[3] Based on spaCy's sentence segmentation: spacy.io

[4] https://fiction.ict.usc.edu/creativehelp/

Initial Story: I knew it wasn't a good idea to put the alligator in the bathtub. The problem was that there was nowhere else waterproof in the house, and Dale was going to be home in twenty minutes.	**Suggested:** I needed to know, too, and I was glad I was feeling it. **Modified:** I needed to know how upset he would be if he found out about my adoption spree.
Initial Story: My brother was a quiet boy. He liked to spend time by himself in his room and away from others. It wasn't such a bad thing, as it allowed him to focus on his more creative side. He would write books, draw comics, and write lyrics for songs that he would learn to play as he got older.	**Suggested:** He'd have to learn to get in touch with my father. **Modified:** He had an ok relationship with my parents, but mostly because they supported his separation.

Table 1: Examples of generated suggestions and corresponding modifications with their preceding context

the degree to which authors edited the suggestions. In particular, we calculated the similarity between each suggestion and corresponding modification in terms of Levenshtein edit distance: $1 - \frac{dist(sug,mod)}{max(|sug|,|mod|)}$, where higher values indicate more similarity. The mean similarity score for this dataset was 0.695 (SD=0.346), indicating that authors most often chose to retain large parts of the suggestions instead of fully rewriting them. We investigated whether these similarity scores could be predicted by the linguistic features of the suggestions. Features that significantly correlate with Levenshtein similarity can be interpreted as being 'helpful' in influencing authors to make use of the original suggestion in their story. It is certainly possible to use other similarity metrics to quantify helpfulness, such as similarity in terms of word embeddings. These measures may model similarity below the surface text of the suggestion, in which the modification may use different words to alternatively express the same story event or idea.

With this approach, given a metric for any feature, the helpfulness of that feature can be quantified. Here, we selected some features used in previous work on story generation and evaluating writing quality. In particular, we included some features used in systems applied to the Story Cloze Test (Mostafazadeh et al., 2016), which involves selecting the most likely ending for a given story from a provided set of candidates. Roemmele et al. (2017a) also explored some of these metrics to compare different models for sentence-based story continuation in an offline framework. Our metrics consist of those that analyze the individual features of a sentence by itself (story-independent, Metrics 1-7 below), and those that analyze the sentence with reference to the story context that precedes the suggestion (story-dependent, Metrics 8-14 below). For the story-dependent metrics, we only considered suggestions that did not appear as the first sentence in the story (910 suggestions).

Sentence Length: The length of a candidate ending in the Story Cloze Test was found to predict its correctness (Bugert et al., 2017; Schwartz et al., 2017). We measured the length of suggestion in terms of its number of words (Metric 1).

Grammaticality: Grammaticality is an obvious feature of high-quality writing. We used Language Tool[5] (Miłkowski, 2010), a rule-based system that detects various grammatical errors. This system computed an overall grammaticality score for each sentence, equal to the proportion of total words in the sentence deemed to be grammatically correct (Metric 2).

Lexical Frequency: Writing quality has been found to correlate with the use of unique words (Burstein and Wolska, 2003; Crossley et al., 2011). We computed the average frequency of the words in each suggestion according to their Good-Turing smoothed counts in the Reddit Comment Corpus[6] (Metric 3).

Syntactic Complexity: Writing quality is also associated with greater syntactic complexity (McNamara et al., 2010; Pitler and Nenkova, 2008). We examined this feature in terms of the number and length of syntactic phrases in the generated sentences. Phrase length was approximated by the number of children under each head verb/noun as given by the dependency parse. We counted the total number of noun phrases (Metric 4) and words per noun phrase (Metric 5), and likewise the number of verb phrases (Metric 6) and words per verb phrase (Metric 7). These metrics were all normalized by sentence length.

Lexical Cohesion: Correct endings in the Story Cloze Test tend to have higher lexical similarity to their contexts according to statistical measures of similarity (Mihaylov and Frank, 2017; Mostafazadeh et al., 2016; Flor and Somasundaran, 2017). We analyzed lexical cohesion be-

[5]Code at: pypi.python.org/pypi/language-check
[6]spacy.io/docs/api/token

tween the context and suggestion in terms of their Jaccard similarity (proportion of overlapping words; Metric 8), GloVe word embeddings[7] trained on the Common Crawl corpus (Metric 9), and sentence (skip-thought) vectors[8] (Kiros et al., 2015) trained on the BookCorpus (Metric 10). For the latter two, the score was the cosine similarity between the means of the context and suggestion vectors, respectively.

Style Consistency: Automated measures of writing style have been used to predict the success of fiction novels (Ganjigunte Ashok et al., 2013). Moreover, Schwartz et al. (2017) found that simple n-gram style features could distinguish between correct and incorrect endings in the Story Cloze Test. We examined the similarity in style between the context and suggestion in terms of their distributions of coarse-grained part-of-speech tags, using the same approach as Ireland and Pennebaker (2010). The similarity between the context c and suggestion s for each POS category was quantified as $1 - \frac{|pos_c - pos_s|}{pos_c + pos_s}$, where pos is the proportion of words with that tag. We averaged the scores across all POS categories (Metric 11). We also looked at style in terms of the Jaccard similarity between the POS trigrams in the context and suggestion (Metric 12).

Sentiment Similarity: The relation between the sentiment of a story and a candidate ending in the Story Cloze Test can be used to predict its correctness (Flor and Somasundaran, 2017; Goel and Singh, 2017; Bugert et al., 2017). We applied sentiment analysis to the context and suggestion using the tool[9] described in Staiano and Guerini (2014), which provides a valence score for 11 emotions. For each emotion, we computed the inverse distance $\frac{1}{(1+|e_c - e_s|)}$ between the context and suggestion scores e_c and e_s, respectively. We averaged these values across all emotions to get one overall sentiment similarity score (Metric 13).

Entity Coreference: Events in stories are linked by common entities (e.g. characters, locations, and objects), so coreference between entity mentions is particularly important for establishing the coherence of a story (Elsner, 2012). We calculated the proportion of entities in each suggestion that coreferred to an entity in the corresponding context[10] (Metric 14).

[7] nlp.stanford.edu/projects/glove
[8] github.com/ryankiros/skip-thoughts
[9] github.com/marcoguerini/DepecheMood/releases
[10] Using CoreNLP: stanfordnlp.github.io/CoreNLP

4 Results and Conclusion

	ρ
1. Sentence length	-0.082
2. Grammaticality	0.097
3. Word frequency	0.058
4. # NPs	0.112
5. NP length	0.052
6. # VPs	0.001
7. VP length	-0.022
8. Jaccard sim	0.017
9. GloVe sim	0.105
10. Skip-thought sim	0.258
11. Word POS sim	-0.037
12. Trigram POS sim	-0.023
13. Sentiment sim	0.107
14. Coreference	0.134

Table 2: Correlation ρ between metric scores for suggestions and similarity to modifications

Table 2 shows the Spearman correlation coefficient (ρ) between the suggestion scores for each metric and their Levenshtein similarity to the resulting modifications. This coefficient indicates the degree to which the corresponding feature predicted authors' modifications, where higher coefficients mean that authors applied fewer edits. Statistically significant correlations ($p < 0.005$) are highlighted in gray, indicating that suggestions with higher scores on these metrics were particularly helpful to authors. Suggestion length did not have a significant impact, but grammaticality emerged as a helpful feature. The frequency scores of the words in the suggestions did not significantly influence their helpfulness. In terms of syntactic complexity, suggestions with more noun phrases were edited less often, but verb complexity was not influential. For lexical cohesion, the number of overlapping words between the suggestion and its context (Jaccard similarity) was not predictive, but vector-based similarity was an indicator of helpfulness. Similarity in terms of sentence (skip-thought) vectors was the most helpful feature overall, which suggests these representations are indeed useful for modeling coherence between neighboring sentences in a story. Analogously, Roemmele et al. (2017b) and Srinivasan et al. (2018) found that these representations were very effective for encoding story sentences in the Story Cloze Test in order to predict

correct endings. Neither metric for style similarity predicted authors' edits, but sentiment similarity between the suggestion and context was significantly helpful. Finally, suggestions that more frequently coreferred to entities introduced in the context were more helpful.

These results describe this particular sample of Creative Help interactions for a selected set of features relevant to story generation, but this analysis can be scaled to determine the influence of any feature in an automated writing support framework where authors can adapt generated content. The objective of this approach is to leverage data from user interactions with the system to establish an automated feedback loop for evaluation, by which features that emerge as helpful can be promoted in future systems.

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

Margaret A Boden. 2004. *The creative mind: Myths and mechanisms*. Psychology Press.

Michael Bugert, Yevgeniy Puzikov, Andreas Rücklé, Judith Eckle-Kohler, Teresa Martin, Eugenio Martínez-Cámara, Daniil Sorokin, Maxime Peyrard, and Iryna Gurevych. 2017. LSDSem 2017: Exploring data generation methods for the story cloze test. In *LSDSem 2017*.

Jill Burstein and Magdalena Wolska. 2003. Toward Evaluation of Writing Style: Finding Overly Repetitive Word Use in Student Essays. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 35–42.

Elizabeth Clark, Anne Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative Writing with a Machine in the Loop: Case Studies on Slogans and Stories. In *Proceedings of the 23rd ACM Conference on Intelligent User Interfaces*. IUI'2018.

Scott A. Crossley, Jennifer L. Weston, Susan T. McLain Sullivan, and Danielle S. McNamara. 2011. The Development of Writing Proficiency as a Function of Grade Level: A Linguistic Analysis. *Written Communication* 28(3):282–311.

Terry Dartnall. 2013. *Artificial intelligence and creativity: An interdisciplinary approach*, volume 17. Springer Science & Business Media.

Micha Elsner. 2012. Character-based Kernels for Novelistic Plot Structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. EACL '12.

Michael Flor and Swapna Somasundaran. 2017. Sentiment Analysis and Lexical Cohesion for the Story Cloze Task. In *LSDSem 2017*.

Vikas Ganjigunte Ashok, Song Feng, and Yejin Choi. 2013. Success with Style: Using Writing Style to Predict the Success of Novels. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, pages 1753–1764.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an Interactive Poetry Generation System. In *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics, pages 43–48.

Pranav Goel and Anil Kumar Singh. 2017. IIT (BHU): System Description for LSDSem17 Shared Task. In *LSDSem 2017*.

Molly E Ireland and James W Pennebaker. 2010. Language style matching in writing: synchrony in essays, correspondence, and poetry. *Journal of personality and social psychology* 99(3):549.

Ahmed Khalifa, Gabriella AB Barros, and Julian Togelius. 2017. DeepTingle. In *International Conference on Computational Creativity 2017*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations*. San Diego.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, pages 3294–3302.

Antonios Liapis, Georgios N Yannakakis, Constantine Alexopoulos, and Phil Lopes. 2016. Can computers foster human users creativity? Theory and praxis of mixed-initiative co-creativity. *Digit. Cult. Educ. (DCE)* 8(2):136–152.

Enrique Manjavacas, Folgert Karsdorp, Ben Burtenshaw, and Mike Kestemont. 2017. Synthetic Literature: Writing Science Fiction in a Co-Creative Process. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)*. pages 29–37.

Neil McIntyre and Mirella Lapata. 2009. Learning to Tell Tales: A Data-driven Approach to Story Generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 217–225.

Danielle S. McNamara, Scott A. Crossley, and Philip M. McCarthy. 2010. Linguistic features of writing quality. *Written Communication* 27(1):57–86.

Todor Mihaylov and Anette Frank. 2017. Story Cloze ending selection baselines and data examination. In *LSDSem 2017*.

Marcin Miłkowski. 2010. Developing an open-source, rule-based proofreading tool. *Softw. Pract. Exper.* 40(7):543–566.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *Proceedings of the 14th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*. pages 839–849.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 311–318.

Rafael Pérez y Pérez and Mike Sharples. 2001. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 186–195.

Melissa Roemmele and Andrew S Gordon. 2015. Creative Help: A Story Writing Assistant. In *International Conference on Interactive Digital Storytelling*. Springer International Publishing.

Melissa Roemmele and Andrew S. Gordon. 2018. Automated Assistance for Creative Writing with an RNN Language Model. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*. ACM, New York, NY, USA, IUI'18, pages 21:1–21:2.

Melissa Roemmele, Andrew S Gordon, and Reid Swanson. 2017a. Evaluating Story Generation Systems Using Automated Linguistic Analyses. In *SIGKDD 2017 Workshop on Machine Learning for Creativity*.

Melissa Roemmele, Sosuke Kobayashi, Naoya Inoue, and Andrew Gordon. 2017b. An RNN-based Binary Classifier for the Story Cloze Test. In *LSDSem 2017*.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the Goals of Grammatical Error Correction: Fluency Instead of Grammaticality. *Transactions of the Association for Computational Linguistics* 4:169–182.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A Smith. 2017. The Effect of Different Writing Tasks on Linguistic Style: A Case Study of the ROC Story Cloze Task. In *The SIGNLL Conference on Computational Natural Language Learning (CoNLL 2017)*.

Siddarth Srinivasan, Richa Arora, and Mark Riedl. 2018. A Simple and Effective Approach to the Story Cloze Test. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2018)*.

Jacopo Staiano and Marco Guerini. 2014. DepecheMood: A lexicon for emotion analysis from crowd-annotated news. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*.

Oliviero Stock and Carlo Strapparava. 2005. The act of creating humorous acronyms. *Applied Artificial Intelligence* 19(2):137–151.

Reid Swanson and Andrew S. Gordon. 2012. Say Anything: Using Textual Case-Based Reasoning to Enable Open-Domain Interactive Storytelling. *ACM Transactions on Interactive Intelligent Systems* 2(3):1–35.

Tony Veale and Yanfen Hao. 2007. Comprehending and generating apt metaphors: a web-driven, case-based approach to figurative language. In *Proceedings of AAAI*.