Something Something Creative Writing

Ben Greene
Galen Berger-Fletcher
Tenzin Dophen
Malcom Grossman

October 9, 2018

1 Vision

There is not a single writer who hasn't dealt with the issue of writer's block. It's a problem that can destroy someone's creative potential for extended periods of time with seemingly little recourse. Writers often try to overcome this hurdle and to generate inspiration for their stories with collaborative writing games, which require the participation of multiple people in order to help break down the participants' creative barriers. Our project seeks to provide an alternative to these cooperative writing games by providing solo writers with an artificial writing partner.

We propose a neural network that will generate inspiration during the creative writing process to help mitigate the effects of or even prevent writer's block (Clark et. al). This neural network, under the working title of WriterBot, will function as an in-the-loop aid to the writer in an interactive system where both parties take turns writing sentences. The writer will write a sentence and WriterBot will generate another sentence based on the writer's input to serve as the next line of the story. The writer will then have a chance to edit or delete WriterBot's sentence before continuing onto their own next sentence, ultimately creating a story that is the product of both machine and human inspiration.

We hope to extend Clark et al.'s human-centric approach by giving users the option to choose when to seek suggestions from the machine (Roemmele et al.) in contrast to suggestions being automatically generated by the machine after each sentence input, thereby giving people the ability to control the level of machine intrusiveness in their writing process. We will also try to generalize the original paper's approach by exploiting information from a larger context outside of just the user's previous sentence, as well as by training the machine with data from different genres of fiction. Finally, we would like to extend the original paper by providing a more robust suite of related tools, including prompt generation and the ability to save and share WriterBot-generated stories. We hope WriterBot will be able to inspire writers to take their stories in new and creative directions, and to provide an enjoyable way to get unstuck in the creative writing process.

2 Description

To generate WriterBot's suggestions, we will use a recurrent neural network based language model (RNN LM). RNNs are very efficient in predicting results in sequential data. They allow information to cycle inside the network for a very long time, allowing access to more contextual information than just the previous sentence to predict the next sentence, and provide output from the entire as input to processing subsequent data.

We will use a simple recurrent neural network architecture (Mikolov et al., 2010), which is relatively easy to implement and train. We will also explore different optimization techniques to improve the performance of the neural network.

3 Roadmap

We have two general phases of work, and we will divide them by term. This term is our neural network phase, and winter term will be our UI/UX and feature phase.

By the end of this term, we want to have a working CLI for our RNN, as well as a useable dataset that fits our needs. We plan on writing the majority of our code in Python, so our first step will be to determine which libraries to use for numerical computation and the neural net. Our larger step will be to decide and implement the inner

structure of the RNN, which we plan to have done by the end of the term. Concurrently, we will get access to data. Unfortunately, we are unable to use the fiction database that the original (paper)[cite] used, so finding decent data and scraping it is a priority, because training will require that data. Our current solution is to use alternative data in our preliminary models as a stand-in while better data is found.

Our winter term agenda is to implement a UI for our CLI. Since our code will be in Python, our plan is to use Django to create a web app. We want this done in the first few weeks of winter term so we can work on further feature development. These features are fleshed out more fully, as is our main goal in the Section 7.

We view the RNN as the more technically difficult task in this project, which is why we are placing very heavy priority on its completion as soon as possible. While not trivial, a UI is something that all of us have some experience with, and will require less technical time than perfecting our language model, which, again, is the core of our project.

4 Division of work

We plan on working together as a full group on the most critical parts of the implementation of our RNN. This neural language model will be the conceptual core of our project, so it is important that every group member has the opportunity to develop a high-level understanding of how it works. However, Malcolm and Tenzin will take the lead on figuring out the lower-level details of the RNN and optimizing its performance. Ben will focus on finding and processing our training data, while Galen will begin developing the front-facing elements of our project like our UI and web framework.

5 Individual tasks

Although we will be working as a group on much of the project, we have each set individual goals for the remainder of Fall Term, in alignment with our general directions as discussed above. These goals are as follows:

- Malcolm: have a RNN that compiles and runs to completion in Tensorflow
- Tenzin: find possible optimization strategies to improve the efficiency of the neural network
- Galen: design a basic web interface for the application using Balsamiq, set up a simple webapp with Django for the RNN to slot into
- Ben: Develop data sources via a webscraper and efficiently manage RNN training.

6 Resources

For our initial training and testing of our neural language model on fairly small subsets of our data, we plan to request a dedicated lab machine in CMC 307. This machine will eventually also allow us to serve our webapp until we agree upon a longer-term solution.

As the computational requirements of the training process become clearer to us over the second half of fall term, we will likely need more processing power than a single iMac can offer. There are two possible ways for us to get this increased power. First, we can work with Mike Tie to gain access to the new research cluster (summer18.dmz.carleton.edu). This option is limited due to the cluster being primarily intended for faculty research, but we may be able to get our tasks in at a lower priority. Second, we plan on using the Jetson TX2 Dev Kit for our more advanced training. It's a machine-learning-specific graphics card developed by Nvidia and will allow us to train more advanced neural networks faster and better than our other options. Malcolm has access to this resource through the St. Olaf - Carleton Engineering team, which has two available for use.

To build mockups for our web UI (as well as for any other platforms we may extend onto), we plan on using Balsamiq, a wireframing program. We should be able to get by with free trial options, but if we end up relying on it more than expected during our UI design process, we might request a month-or-two-long subscription to its cloud service (at $9 per month).

7 Deliverables

7.1 Minimumm Viable Product

Our most basic possible deliverable will be a webapp with a simple interface that:

1. Prompts the user with an image
2. Takes input from the user one sentence at a time
3. After each sentence of input:
   (a) Generates a reasonable next sentence using a context-sensitive neural language model
   (b) Displays it to the user for review and editing

7.2 Extensions

Below are a selection of possible improvements to our minimum deliverable as described above, ranging from trivial to wildly optimistic. Depending on how closely we are able to follow our roadmap, we expect to implement at least 2-4 of these.

7.2.1 Core Improvements

- Modern, professional-looking interface design
- Variable context-sensitivity in the language model, so it can take more than the previous sentence into account when generating the next sentence
- Additional language models trained on different genres
- More in-depth writing prompt generator (possibly using our language model or the beam search approach[2])
- Opt-in account system to save users' stories and preferences (and to facilitate the implementation of several additional features below)

7.2.2 Extended Functionality

- Export a completed story in multiple formats
- Save a story in progress and come back to it later
- Share a completed story on social media
- Option to choose when to ask AI for a suggestion rather than automatically getting one after every sentence

7.2.3 Extra Modes

- Multiplayer mode to allow collaboration with another human rather than the AI
- Timed mode to help break down writer's block
- AI mode to watch multiple AI collaborate on a story without human input (for entertainment value)

7.2.4 Additional Platforms

- iOS
- Android
- Desktop app(MacOS or Windows)

References

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model, 2003.

[2] E. Clark, A. S. Ross, C. Tan, Y. Ji, and N. A. Smith. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 329–340, New York, NY, USA, 2018. ACM.

[3] M. Ghazvininejad, X. Shi, Y. Choi, and K. Knight. Generating topical poetry. In *Proceedings of ACL*, pages 1183–1191, 2016.

[4] Y. Ji, T. Cohn, L. Kong, C. Dyer, and J. Eisenstein. Document context language models. In *Proceedings of the 4th International Conference on Learning Representations (Workshop Track)*, 2016.

[5] A. Karpathy. The unreasonable effectiveness of recurrent neural networks. `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`, 2015.

[6] J. Kim, J. Cheng, and M. S. Bernstein. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In *Proceedings of CSCW*, 2014.

[7] T. Mikolov, M. Karafát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Proceedings of Interspeech*, 2010.

[8] M. Roemmele and A. S. Gordon. Creative help: A story writing assistant. In *International Conference on Interactive Digital Storytelling*, page 81–92, 2015.

[9] M. Roemmele and A. S. Gordon. An encoder-decoder approach to predicting causal relations in stories. In *Storytelling Workshop at the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018)*, June 5, 2018.

[10] R. Sloan. Writing with the machine. `https://www.robinsloan.com/notes/writing-with-the-machine/`, 2016.

[11] A. Sordon, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan. A neural netowrk approach to context-sensitive generation of conversational responses. In *Proceedings of ACL*, page 196–205, 2015.

[12] T. Tran. Creating a text generator using recurrent neural network. `https://chunml.github.io/ChunML.github.io/project/Creating-Text-Generator-Using-Recurrent-Neural-Network/`, 2016.

[13] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: a neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.