Writing Stories with Help from Recurrent Neural Networks

Melissa Roemmele
Institute for Creative Technologies
University of Southern California
12015 Waterfront Dr., Los Angeles, CA 90094
roemmele@ict.usc.edu

Data-driven Narrative Intelligence

Automated story generation has a long history of pursuit in artificial intelligence. Early approaches used hand-authored formal models of a particular story-world domain to generate narratives pertaining to that domain (Klein, Aeschlimann, and Balsiger 1973; Lebowitz 1985; Meehan 1977). With the advent of machine learning, more recent work has explored how to construct narrative models automatically from story corpora (Li et al. 2013; McIntyre and Lapata 2009; Swanson and Gordon 2012). This research has created a new potential for interactivity in narrative generation. Unlike previous approaches which lacked the breadth of knowledge required for open-domain storytelling, these systems leverage story data to interface with authors pursuing diverse narrative content. For example, Swanson and Gordon (Swanson and Gordon 2012) demonstrated an application where a user and automated agent took turns contributing sentences to a story. Their system used a case-based reasoning approach to retrieve a relevant continuation of the user's sentence from a large database of stories. This research has given rise to a new type of story generation task, one of "narrative auto-completion", where a system analyzes an ongoing narrative and generates a new contribution to the story. Analogous to existing automated writing aids like spelling and grammar correction, narrative auto-completion is applicable as a writing tool that suggests new ideas to authors.

Recurrent Neural Networks (RNN) are a promising machine learning framework for language generation tasks. In natural language processing (NLP) tasks, RNNs are trained on sequences of text to model the conditional probability distribution of predicting a sequence unit (often a character or word) given the sequence up to that point. After training it is straightforward to generate new text by iteratively predicting the next unit based on the text generated so far. In this same way, a given text can be extended by predicting additional text in the sequence. For this reason an RNN is a suitable engine for an automated story writing assistant that takes an ongoing story as input for predicting a continuation of the story. In this thesis I explore the use of RNNs for this novel generation task, and show how this task affords a unique opportunity for the evaluation of generation systems.

Language Generation with RNNs

RNNs are extremely powerful for NLP tasks, having demonstrated success on tasks like speech recognition (Graves and Jaitly 2014) and machine translation (Sundermeyer et al. 2014). Mikolov et al. (Mikolov et al. 2010) showed that RNNs encode more accurate language models than traditional n-gram statistics, as measured by performance on a standard speech recognition task. The simplest RNN architecture has an input layer, hidden layer, and output layer connected respectively by three weight matrices, W_{in}, W_{hh}, and W_{out}. When the RNN is used as a language model, a dictionary of size *D* associates each known word type with an index. Each word is represented as zero vector of dimensionality *D* with a 1 at the dictionary index of the word type. The input layer encodes the current word, the hidden layer encodes the current underlying state of the sentence, and the output layer encodes the probability distribution for the next word in the sentence. During training, for each timestep *t* in a sentence, the RNN computes the following recurrence:

$$hidden_t = tanh(word_t W_{in} + hidden_{t-1} W_{hh})$$

$$output_t = hidden_t W_{out}$$

Applying a softmax classifier to the output layer gives the probability distribution of the next word over all dictionary words. Training occurs by minimizing a cost function, defined as the negative log-likelihood of the probability of each actual word in the training sequence. The gradient of this cost is then back-propagated in order to update the weight matrices (Rumelhart, Hinton, and Williams 1988).

During generation, the learned probability distribution can be sampled in order to generate a sentence, with each predicted word in the output layer at time *t* being fed back into the input layer at time *t* + 1. The sequential units that RNNs predict are not required to be words; distributions over single characters are analogous, and have actually been favored because of their low dimensionality compared with words. RNNs have been used to generate Wikipedia articles (Sutskever, Martens, and Hinton 2011) and conversational responses (Sordoni et al. 2015). The former work also showed that these models could generate plausible completions to the beginning of sentences. My work focuses on performing this completion task specifically with narrative text.

Automated Story Writing Assistance

This thesis envisions the task of narrative auto-completion applied to helping an author write a story. My advisor and I have built an application called *Creative Help* that provides this type of automated writing assistance (Roemmele and Gordon 2015). In this application, a user writes a story and when she types `\help\`, the application returns a suggested next sentence which the user can edit like any other text in the story. Creative Help tracks modifications to suggestions as a means of evaluating their quality. Performance is quantified as the rate at which users edit suggestions, based on the assumption that suggestions with fewer edits (including deletions) are perceived as better continuations of the story. This functionality addresses a existing weakness in language generation research: unlike classification tasks, there is no single gold standard with which to compare generated output, making it difficult to quantify system performance. Creative Help offers a new paradigm for evaluating language generation organically through user interaction with the application, avoiding the need to conduct evaluation separately from generation.

The previous approach to generation in Creative Help used information retrieval methods to find stories similar to the user's story among a large corpus, and then extract sentences from these stories as suggested continuations. While this approach often generated relevant suggestions, the suggestions modeled the context of their originating story rather than the user's story, limiting their compatibility. RNNs make context-sensitive predictions from aggregated data and thus are more likely to avoid this problem of poorly adapted content.

The goal of my thesis is to implement an RNN-based system for story generation applied to the context of assistive story writing that I have described. I plan to use the evaluation functionality of Creative Help to compare RNNs to alternative approaches such as the described case-based reasoning method. Additional research questions likely to arise from this work are the following: is the word-based language model sufficient for modeling narrative, or are there other structures that should be explicitly modeled (e.g. clauses, sentences, paragraphs)? Is it possible to capture abstract narrative features like plot, theme, and character development by modeling low-level structural units? Can RNNs be used to not just predict narrative text in sequential order but also in the reverse direction (i.e. predict text that occurs before a particular part of the story)? The first step in precisely defining such research questions is to implement an RNN-based generation framework as it is described here. The evaluation of this framework will reveal the specific challenges that must be resolved in modeling narrative with RNNs. At the current time (September 2015), two requirements of this thesis have already been fulfilled: a dataset of 20 million stories has been prepared, and the story writing assistant (Creative Help) interface has been built. By February 2016, I plan to have completed training the initial RNN model and begun experiments using the writing assistant to evaluate the model. Further work after this will examine potential solutions to the most important problems associated with RNN-based narrative generation.

References

[Graves and Jaitly 2014] Graves, A., and Jaitly, N. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning*, 1764–1772.

[Klein, Aeschlimann, and Balsiger 1973] Klein, S.; Aeschlimann, J.; and Balsiger, D. 1973. Automatic novel writing: A status report. *Wisconsin University*.

[Lebowitz 1985] Lebowitz, M. 1985. Story-telling as planning and learning. *Poetics* 14(6):483–502.

[Li et al. 2013] Li, B.; Lee-Urban, S.; Johnston, G.; and Riedl, M. 2013. Story Generation with Crowdsourced Plot Graphs. In *27th AAAI Conference on Artificial Intelligence*.

[McIntyre and Lapata 2009] McIntyre, N., and Lapata, M. 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*, 217–225. Stroudsburg, PA, USA: Association for Computational Linguistics.

[Meehan 1977] Meehan, J. R. 1977. TALE-SPIN, An Interactive Program that Writes Stories. In *5th International Joint Conference on Artificial Intelligence*, 91–98.

[Mikolov et al. 2010] Mikolov, T.; Karafiat, M.; Burget, L.; Cernocky, J.; and Khudanpur, S. 2010. Recurrent Neural Network based Language Model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, 1045–1048.

[Roemmele and Gordon 2015] Roemmele, M., and Gordon, A. S. 2015. Creative help: A story writing assistant. In *International Conference on Interactive Digital Storytelling*. Springer International Publishing.

[Rumelhart, Hinton, and Williams 1988] Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5:3.

[Sordoni et al. 2015] Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.-Y.; Gao, J.; and Dolan, B. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *Proceedings of NAACL-HLT*, 196–205.

[Sundermeyer et al. 2014] Sundermeyer, M.; Alkhouli, T.; Wuebker, J.; and Ney, H. 2014. Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, 14–25.

[Sutskever, Martens, and Hinton 2011] Sutskever, I.; Martens, J.; and Hinton, G. E. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning*, 1017–1024.

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