Object To Story

Jimmy Johnson & Ignacio Marroquin
Texas A&M University-Corpus Christi
Corpus Christi, USA

I. Introduction

Our project is to take an image as input and place it into a generated story. Initially we set out to combine a deep convolutional neural network(DCNN) [1], [2] as the image classifier, and an recurrent neural network (RNN) [3]–[8] to generate stories based on the image classified in the first stage. Additionally the RNN would learn how to represent entities or characters throughout the generated text [5], [7]. However this proved to be a much more difficult problem to approach. Upon realizing this problem we decided to deconstruct the problem further, turning it into a problem we could solve in post processing rather than it being solved by the RNN entirely.

II. Methodology

To approach this problem we require two networks, one CNN to serve as our image classifier, and a RNN to serve as our text sequence generator. After an image is fed into the CNN and a label is returned, we then generate a story using the RNN. Once these two variables are generated the top label prediction from the CNN and the story generated by the RNN is given to the post processor. We use the Natural Language Processing Tool Kit(NLTK) [9] to generate the usage and structure of each word. This provides us what word is an entity or not. Using this information we can replace the found entity with an entity that represents the top label prediction generated by the CNN.

A. Image Classifier

While selecting an image classifier we needed to keep in mind the magnitude of the classifier as a whole. Typically classifiers that reach 70% or higher are usually very deep networks. That being said we first looked at using ResNet [2] which is based on residual blocks, and VGG16 [1] which is also a very widely popular deep convolutional neural network. Currently there are more pre-trained networks available for the VGG16 which caused us it as our image classifier. Figure 1 displays the architecture of VGG16 which follows an encoding pathway paired with fully connected layers to make the final prediction.

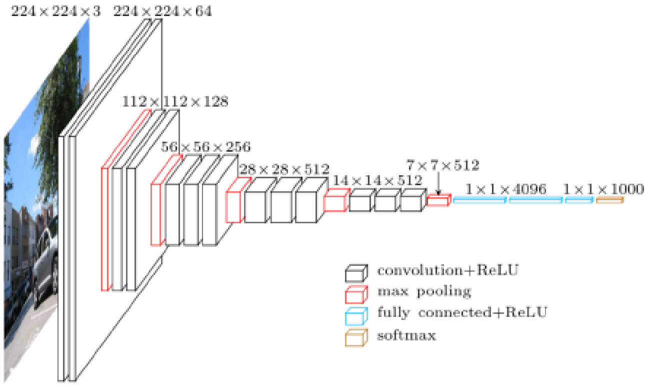
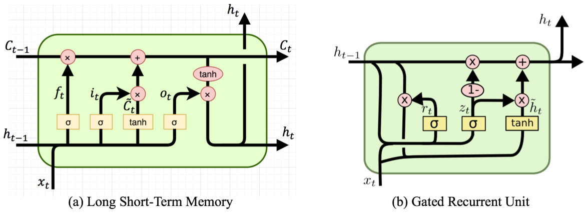


Fig. 1. VGG16's architecture [1].

B. Text Generator

After viewing several RNN's that were based on dynamic entity representation [5], [7], we found that even though we would not be utilizing a framework similar to this, we could take away that using a long short term memory(LSTM) [10] which can maintain a memory of previous states, retaining information that is more important and forgetting information that is less important or important only to the next few states versus the previous say one hundred states. There are several types of LSTM cells to use in an RNN, but the most popular cell type to use for sparser datasets is the Gated Recurrent Unit(GRU) [8] which is It combines the forget and input gates into a single update gate. It also merges the cell state and hidden state, and makes some other changes [8]. Figure 2 compares the vanilla LSTM cell [10] to the GRU cell [8]. The GRU LSTM that we use is composed of five layers and has an internal size of 512.



(a) Long Short-Term Memory
(b) Gated Recurrent Unit

Fig. 2. GRU cell architecture [8].

C. Combining The Two

Figure 3 shows how we combine the two networks with the post processing module. This post processor works by utilizing the NLTK [9] package to label what word is actually an entity. Knowing this information we select a name at random and at the first instance of an entity we add the class type. For example: the first instance of the entity Kyle, would be marked as Kyle the mink if the detected class is in fact a mink. This is done for every entity detected.

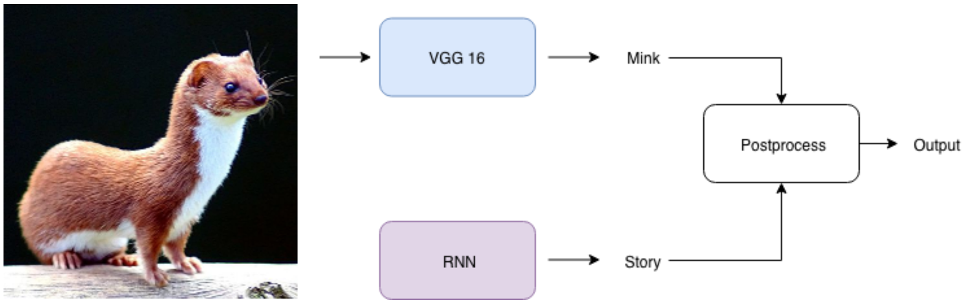


Fig. 3. Displaying the full flow of how a story is generated. From the CNN image classifier, the RNN story generator, and the post processor that puts the label result into our generated story.

III. Interface

After training is complete we include a GUI interface to make running several tests simpler. We include loading different images as well as utilizing a camera if available. Users also have the choice to change the author that is used to generate text as well. We also included a voice component that helps use manage the current tests being run, as well as read the text that has been generated outload.

IV. Experimental Results

Training the RNN was the most difficult area of our project. Selecting a good dataset also affected the results of the post processing module as well. Using a kaggle dataset containing classical literature [11] that includes several authors. Selecting a specific author increases the performance of our RNN as well due to maintaining the writing style of only one individual. The RNN is trained for 50 epochs and has a batch size or sequence length of 50 on Arthur Conan Doyle due to the amount of characters that he uses. Additionally we also trained on Aristotle, and Edgar Allen Poe.

VGG16 ImageNet Top 5	92.7%
GRU LSTM Final Test	70.2%

The table above displays the accuracy score for both networks. VGG16's pre-trained model claims to achieve 92.7% accuracy [1]. Our RNN achieved 70.2% accuracy after training completed.

A. Example output

Some example output is shown that is unedited:
Lady Ledah the mink – that is the last three men, who have been the secret of the statesman who has been an actual presence." " Alikee the mink, then, what do you make of it, Mr. Marcellina the mink, anyhow?" " I think I should have the present if it went down the rest. It was a common serious message, and that the document was a considerable adventure of the servants. It was the second thing that I had seen the man who had been sent to me. It was the more sensitive than I could see that his eyes were standing by a constant late and silent research which had been seen before the crime. He had been the sea and the bed and the creak of the house. There was a strange country stream of steps, and then suddenly sprang to the window and stood behind the door. Marcellina was so shaken with a pair of delight which had allowed him to be a most universal time.

V. Discussion

The output that we generate only contains very small spelling errors. As a whole the text makes enough sense to draw up where a story is going. Replacing the characters that are originally generated seems to change the tone of the story depending on what type of class it is switched out for. Using our imagination also helps make the story more lively, for example if the predicted label is a dog, we tend to picture dogs attempting to solve a crime.

VI. Conclusion

As a whole the project we initially set out to complete turned to be ambitious but our edited goal was reached with results that showed our concept did work well. Moving forward we plan to add more authors to our model, and also optimize the flow of the deploy operations.

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[3] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[4] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," 2018.

[5] E. Clark, Y. Ji, and N. A. Smith, "Neural text generation in stories using entity representations as context," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, vol. 1, 2018, pp. 2250–2260.

[6] I. V. Serban, T. Klinger, G. Tesauro, K. Talamadupula, B. Zhou, Y. Bengio, and A. C. Courville, "Multiresolution recurrent neural networks: An application to dialogue response generation." in *AAAI*, 2017, pp. 3288–3294.

[7] Y. Ji, C. Tan, S. Martschat, Y. Choi, and N. A. Smith, "Dynamic entity representations in neural language models," *arXiv preprint arXiv:1708.00781*, 2017.

[8] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[9] S. Bird and E. Loper, “Nltk: the natural language toolkit,” in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2004, p. 31.

[10] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] M. O’Neill. (2017) Classic literature in ascii. [Online]. Available: https://www.kaggle.com/mylesoneill/classic-literature-in-ascii