Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

From Quick-draw To Story: A Story Generation System for Kids' Robot

Lecheng Wang*, Shizheng Qin*, Menglong Xu*, Rui Zhang*, Lizhe Qi† and Wenqiang Zhang*
Shanghai Key Laboratory of Intelligent Information Processing
* *School of computer Science*
† *Academy for Engineering & Technology*
Fudan University
Shanghai, P.R.China
{*wanglc17, szqin17, mlxu17, zhangrui, qilizhe, wqzhang*}@*fudan.edu.cn*

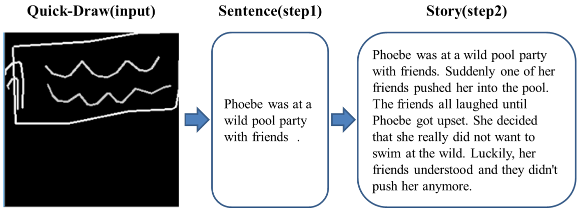
***Abstract*— From quick-draw to story aims to draw a narrative story based on simple lines children draw. On account of the simple knowledge contained in quick-draw and lack of labelled data, this task is assuredly challenging. In this paper, it is divided into two subtasks. Firstly, a Multitask Transformer Network is proposed to generate the sentence on the grounds of the information of quick-draw. Secondly, we finetune the OpenAI Generative Pre-training Model (OpenAI GPT) to generate a story on the basis of the sentence. However, evaluating the text generation model is difficult. In order to solve with this, a new evaluation metric based on Story Cloze Test is proposed. Our evaluations indicate the prominent results our method has achieved in this task.**

***Index Terms*— Robot, Image Caption, Story Generation, Transformer, Pre-training Model**

I. Introduction

Quick Draw[1] is an online game developed by Google that to enable players to draw a picture of an object, after which the system would tell the players what the figure represents. This game is very suitable to be a functional model of the robots that we are developing to accompany children to grow up. More feedback than a few simple possible classification results is expected in case of applying the Google Quick Draw game to the robot for kids. Therefore to design a new task aiming to generate high-quality short stories with our model to interact with children based on the quick-draw images input by themselves.

To some extent, the task can be regarded as an image caption task[27], [10], [28], [2]. The encoder is used to obtain high-level information of the image, according to which the decoder is designed to generate a smooth and well-structured sentence. But unlike the traditional image caption task, quick-draw tends to have simple lines and often only provides simple content information. As a result, some methods[3], [14] to improve the quality of image caption are not effective in our task, such as combining attention mechanism on multi-level feature maps, adding high-level semantic attributes of images to encoder-decoder structures.



Phoebe was at a wild pool party with friends .

Phoebe was at a wild pool party with friends. Suddenly one of her friends pushed her into the pool. The friends all laughed until Phoebe got upset. She decided that she really did not want to swim at the wild. Luckily, her friends understood and they didn't push her anymore.

Fig. 1. Our model generates the sentence based on quick-draw and then generates the whole story based on the generated sentence.

Automated stories use a series of coherent words to describe some reasonable events[22], [9], [15]. Lacking labelled data, generating stories based on existing fragments or topics is difficult, so we divide the encoding process into two subtasks as Figure 1 shows: 1) sentence generation task: decoder generates the first sentence of the story based on the image information, and decides the tone. 2) story generation task: decoder continues to write the whole story based on the previously generated sentence.

In this paper, we combine the images in Quick Draw with the corpus in ROCStories[16]. After extracting all the sentences in ROCStories and associating them with the quick-draw for the first step of training, we use the ROCStories corpus for the second step to predict the whole story based on the first sentence. The Multi-task Transformer Network we propose uses resnet18[8] as encoder to capture high-level features of images. Inspired by [31], we add a classification task to our model to help tell the object of quick-draw. With the help of pre-trained language model, a lot of NLP tasks have achieved good results. Therefore to use OpenAI GPT[20] as a pre-trained model for decoder.

A significant challenge for our task is that there is no reasonable and objective evaluation of the story generated by the model. In order to solve this problem, we have developed an evaluation method based on Story Cloze Test[17], which can assess the quality of the generated stories objectively. What Story Cloze Test requires for the model is to understand the story from the content and select the correct ending of

[1] `https://quickdraw.withgoogle.com/`

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

the story in two sentences. In the story generation task, a generative model is employed to generate the ending of the story as per the content and replace the wrong sentence part of the test data of ROCStories Cloze Test with correct ones. Then we shall observe the performance of the outstanding model in the ROCStories Cloze Test in the new test data , in order to evaluate its effectiveness to generate stories.

II. Related Work

A. Image Caption

The Image Caption task is to translate an image into a suitable sentence. [27] used CNN as the encoder and LSTM as the decoder to give the training image a maximum likelihood probability of its corresponding target description statement. [28] used a network structure similar to [27] and introduced an attention mechanism, arguing that the decoder's focus on the image at different moments is different. [31] proposed a Multi-task Learning Approach for image captioning to improve the quality of the generated description. Different from previous works, we use the transformer block[25] as our decoder to generate the sentence.

B. Story Generation

Text generation is an important research area in natural language processing. [22], [9] have proved that the deep neural network can capture the content and plot of the story, on the basis of which to generate the story. Researchers[24], [12] have explored many ways to generate stories by writing a plan. Inspired by [29], [7], we turn the task into two stages: getting a sentence to set the tone for the story, and then generating a story based on the sentence.[19], [20], [6] have proved that it works well for the steps to conduct the pre-trained for the language model on a variety of unlabeled corpora firstly and then to finetune specific tasks.

C. Generation Evaluation

The image caption task is typically evaluated using BLEU[18], METEOR[5], ROUGE[13] and CIDEr[26]. [30] believes that there is currently no objective and comprehensive evaluation of the natural language generated by the generative model. The existing evaluation methods, such as BLEU are hard to evaluate the semantic similarity of different sentences. Inspired by [6], [30] proposed BERTSCORE to compute similarity using contextualized BERT embeddings.

III. Methods

Begin with in formulating our problem. Input a quick-draw, the system can generate a sentence based on the image information. The topic of the image description is related to the object in the quick-draw, and then the system can generate a small story according to the image description. Given an quick-draw x, for target classification problem, we use the one-hot vector $y^a = \{y_0^a, y_1^a, ..., y_c^a\}$ to represent the category of the target. y_i^a is 1 if the target x belongs to i-th class with others to be 0, and c represents the number of categories. For the image caption problem, we use the sentences $y^t = \{y_0^t, y_1^t, ..., y_m^t\}$ in the ROCStories which contains the object of quick-draw as ground truth. For the story generation problem, we take the first sentence of the story in ROCStories as input and the part as ground truth.

Our framework is illustrated in Figure 2. We use traditional encoder-decoder as the overall structure of the model. We train the relevant classification task while training our task to generate a story based on quick-draw. Our encoder model shares its CNN network with the classification task to improve the feature extraction capability. In the choice of decoder model, we abandon the traditional LSTM[23] and GRU[4] structure and select the transformer block[25]. The main reasons are as follows:1) With simple line information containing for the quick-draw, excessive attention to image features will reduce the ability of the model. 2) Many pre-trained models based on the transformer have achieved excellent results in NLP tasks. In sentence generation task, we use the model structure of [25], mapping the image features to different words related to the object as the key and value of the multi-head attention. In the story generation task, we abandon the image features and fine-tune the OpenAI GPT pre-trained model to generate the story directly.

A. Feature Extraction

We use ResNet-18[8] to extract image features. Model that is pre-trained on ImageNet is used for experimental comparison. We extract the 512-dimensional features of the final fc layer output of ResNet-18 as image features to map our image x onto vector L. The image features are used as a classification task through a linear layer, and the cross-entropy loss of the classification task is calculated as part of the final loss function. Classification loss is calculated as:

$$z = CNN(x) = \{z_0, z_1, ..., z_l\} \tag{1}$$

$$s = softmax\left(W_a z + b_a\right) \tag{2}$$

$$l_a = -\sum_{j=0}^{n} y_i^a log s_i \tag{3}$$

where z represents the image feature, and s represents the result of classification.

B. Sentence Generation

Inspired by [25], we use a cascade of six layers of transformer blocks to generate image descriptions. The input of the model is divided into two parts, one of which is the text that has been generated. For a quick-draw, we have its corresponding description $y^t = \left\{y_0^t, y_1^t, ..., y_{m-1}^t\right\}$, where m represents the maximum length of the sentence we set. We first add start token to the beginning of the sentence, and then all the sentences will be padded to fit

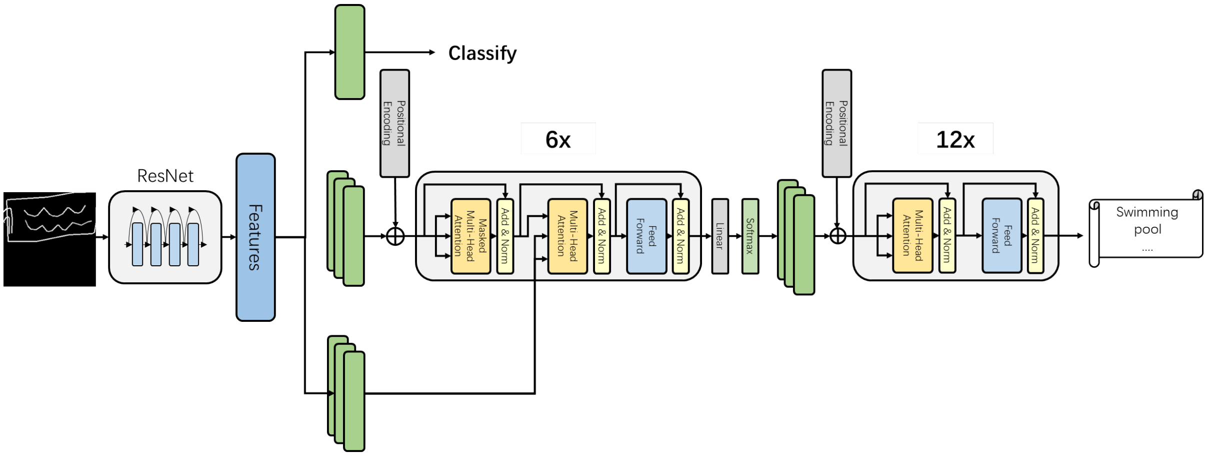


Fig. 2. The Architecture of model

the maximum length $x^t = \left\{ \langle SOS \rangle, y_0^t, y_1^t, ..., y_{m-1}^t \right\}$. First of all, we map the input sentences into the word vector space. Unlike CNN or RNN models, which have position or timing differences, Transformer needs to add position information to the input. We use sinusoidal position encoding[25] to encode the position information and add it to the word vector as input to the model. After that, multihead attention is performed that multiple self-attentions are used to learn the multiple sets of the relationship between the current word and the above:

$$c^t = WordEmb\left(x^t\right) \ + PosEmb\left(x^t\right) \tag{4}$$

$$Attention\left(Q, K, V\right) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

$$MultiHead\left(Q, K, V\right) = Concat(head_1, ...)W \tag{6}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{7}$$

$$h^t = MultiHead\left(c^t, c^t, c^t\right) \tag{8}$$

$$o^t = LayerNorm\left(c^t + h^t\right) \tag{9}$$

where c^t represents the vector of the text that has been generated, h^t contains multiple relationships of words above, and o^t learns the residual information to improve the performance of the network. Layer Normalization is used to accelerate the training process.

The other part is the feature maps of the image. We have obtained the image features corresponding to quick-draw in the image extraction stage. We map the image features linearly to multiple word vector spaces, which we believe will result in a series of words related to the image theme. Using the previous information o^t above, we use multihead attention to learn multiple weights of these words, combine which with the above information o^t as input to the Feed Forward layer. The model will capture image features by this way:

$$c^\theta = f_\theta\left(z\right) \tag{10}$$

$$h^\theta = MultiHead\left(o^t, c^\theta, c^\theta\right) \tag{11}$$

$$o^\theta = LayerNorm\left(o^t + h^\theta\right) \tag{12}$$

here f_θ maps the image features to word vectors related to the image object. Then, o^t is used to learn the weight of these word vectors and add h^θ to output.

The Feed Forward layer does not consider the relationship between the word at the current location and other locations. It does a reorganization of the channel for each location input and adds non-linear capabilities to the model. We calculate the cross-entropy between the generated word currently and label, and the final loss is calculated as follows:

$$FFN\left(x\right) = max\left(0, xW_1 + b_1\right)W_2 + b_2 \tag{13}$$

$$p = f\left(LayerNorm(o^\theta + FFN\left(o^\theta\right))\right) \tag{14}$$

$$l_t = -\sum_{j=0}^{m-1} y_j^t log p_j \tag{15}$$

$$L = l_t + \alpha l_a \tag{16}$$

where FFN means the Feed Forward layer and p represents the result of the generated word currently. α is a hyper-parameter to balance two losses.

C. Story Generation

Story generation model is illustrated in Figure 3. The GPT [20] model uses a 12-layer unidirectional Transformer structure, using over 7,000 books in the BooksCorpus dataset to pre-train the language model, achieving a model with a strong understanding of natural language, a great deal of world knowledge and the ability to deal with long-term dependencies. We fine-tune the GPT pre-trained model on the

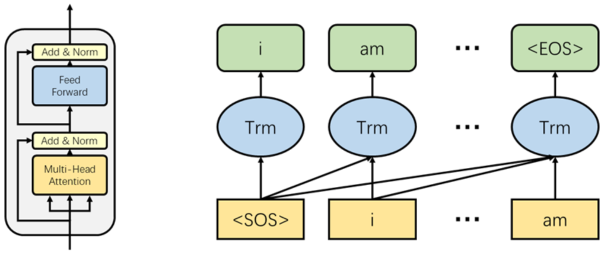


Fig. 3. The story generation model

ROCStories corpus, using the first sentence of the story as input, and the rest as the output of the model. Compared with the sentence generation model, with self-attention used only, the GPT model can entirely focus on the above information of the story when it predicts at the current position. In the end, we get a decoder with strong generalisation ability to generate related stories based on one sentence.

IV. Experimental Setup

A. Dataset

For the sentence generation task, we use the Google Quick Draw dataset and ROCStories corpus[16] as training data. The Quick Draw dataset is a collection of 50 million drawings, which is divided into 345 categories. ROCStories corpus contains 52,665 short stories as training data and 1817 as test data. Each story consists of 5 short sentences that describe daily life. We extract all sentences from the ROCStories corpus training set into a new collection. To map the quick-draw to the sentence, we think the sentence shall be associated with the label if the sentence contains a quick-draw category. To keep the balance of the data, we only keep 134 categories associated with more than 50 sentences. We randomly group the quick-draw and category-associated sentence. For each label, we randomly select 1000 groups for training, 100 for validation and 100 for testing. For the story generation task, we use the whole ROCStories corpus to train and test.

B. Baseline Methods

For the sentence generation task, we compare our method with some traditional image caption methods: GRU[4]; NIC[27]; ATTN[28]; FC[21]. As the story generation task, we use the traditional Seq2seq model[1] as baseline.

C. Hyper-parameters

We use the OpenAi pre-trained tokenizer in a unified way with a vocabulary size of 40480, including start token $\langle SOS \rangle$ and end token $\langle EOS \rangle$.For the sentence generation task, we use Adam optimizer[11] with $\beta_1 = 0.9$,$\beta_2 = 0.98$ and $\epsilon = 10^{-9}$.We try to ensure that the number of all model parameters is the same. For the RNN model, we use 256 for hidden size. For the transformer model, we use six transformer blocks. The number of heads in the multihead attention layer is 8 for each block and the size of query, key and value in attention is 32. We set the dropout to 0.2 and the word vector dimension to 256 uniformly. Besides, we set the α to 1 as the weight of the loss of classification task. For the story generation task, we use the Adam optimizer with a learning rate of 6.25e-5 for fine-tuning.

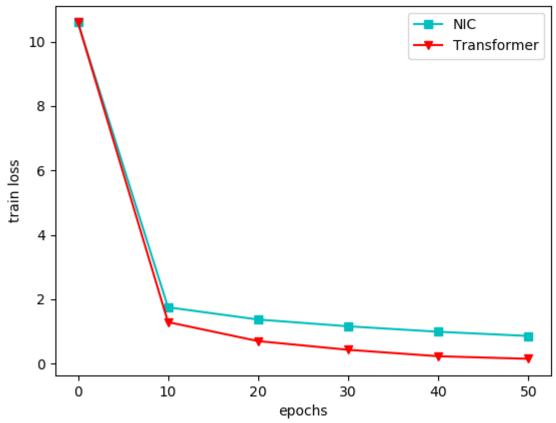


Fig. 4. The process of loss reduction of RNN model and Transformer model during training.

D. Evaluation

For the sentence generation task,we use BLEU-N (N=1,2,3,4) to evaluate the fluency of sentences, and BERTSCORE to evaluate the rationality of sentences (whether it is related to quick-draw). For the story generation task, we collected some discriminators of the ROCStories cloze test to evaluate our generative model.

V. Results and Discussion

A. Sentence Generation

In the traditional image caption task, the image contains rich semantic information, and there are multiple translation methods for the same image. Therefore, the test method is based on an image generated sentence and multiple reference sentences for evaluation. In our situation, the image information provided by quick-draw is not strong enough, and the model can only determine the subject of the sentence based on the image, so the test method of a pair of images corresponding to multiple reference sentences is not objective in our task. Therefore, we apply the Semi-teacher forcing test method. At the time of testing, we give the model a start sequence from the reference sentence, based on which the model generates a sentence. We compare the BLEU and BERTSCORE between the generated sentence and the reference sentence corresponding to the start sequence to evaluate our model.

TABLE I

THE TEST RESULTS OF DIFFERENT MODELS UNDER OUR TEST METHOD.BLEUSCORE EVALUATES THE FLUENCY OF GENERATED SENTENCES, AND BERTSCORE EVALUATES THE SEMANTIC SIMILARITY OF SENTENCES. PP REPRESENTS THE PERPLEXITY. THE BEST SCORE PER EVALUATION METHOD IS SHOWN IN BOLD.

Method	BLEU1	BLEU2	BLEU3	BLEU4	P_{BERT}	R_{BERT}	F_{BERT}	PP
GRU	0.685	0.557	0.489	0.430	-	-	-	2.82
NIC	0.703	0.588	0.524	0.467	0.798	0.787	0.792	2.67
ATTN	0.713	0.592	0.527	0.473	-	-	-	2.48
FC	0.701	0.576	0.505	0.447	-	-	-	2.60
Transformer	0.755	0.654	0.593	0.539	0.944	0.941	0.942	2.14
P-Transformer	0.741	0.632	0.568	0.513	0.939	0.935	0.936	2.25
M-Transformer	**0.767**	**0.679**	**0.625**	**0.573**	**0.953**	**0.951**	**0.952**	**2.09**

TABLE II

EXAMPLE STORY GENERATED BY SEQ2SEQ AND OUR MODEL RESPECTIVELY BASED ON THE SENTENCE.

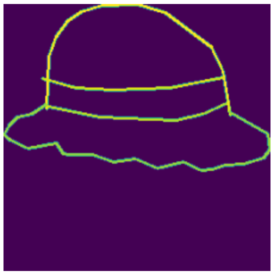
	**Sentence**:the hat she wanted was too expensive.	**Story(Seq2seq)**:the hat she wanted was too expensive. the she'd bought her dad's attention. it was a bit too hard! she bought the supplies she could find. it was too hard!

Table I shows the effectiveness of different models to generate a sentence for a quick-draw input under our test method. For the RNN model, we provide the decoder with convolutional layer information with the attention mechanism for the encoder stage. We find that the effectiveness of the model to generate sentences has not improved. For the Transformer model, we find that using ImageNet pre-trained models (P-Transformer) does not improve the capabilities of the model. Therefore, for our task, we expect our model to pay more attention to the above information than the image information. Figure 4 shows the loss of the Transformer model and the RNN model during training. The Transformer model can converge faster than the RNN model and get a better solution. We added the relevant classification task while generating the sentences. The classification tasks can help our model to focus on the image subject. The Transformer model with the related classification task (M-Transformer) achieves the best results in our tests.

TABLE III

THE RESULTS OF EVALUATING THE STORY GENERATION MODEL BY ROCSTORIES CLOZE TEST.

Method	B-score	N-score	P-score
Seq2seq(Bi-LSTM)	0.68	0.68	0.43
Ours(Bi-LSTM)	0.68	0.49	0.61
Ours(GPT)	0.87	0.53	-

B. *Story Generation*

We use the discriminator from the ROCStories Cloze Test task to evaluate our model. B-score represents the accuracy of the discriminant to identify the end of the story. Our model can generate the end of the story based on the sentences above. N-score represents the accuracy of the discriminator when we replace the negative samples in the test data with the ending which our model generates while P-score represents that we replace the positive samples.

We use a Bi-LSTM[2] based weak discriminator to evaluate the quality of the generated story by the Seq2seq and our model respectively. Table III shows the effectiveness of different models to generate stories, and Table II shows a story generated by different models. It can be found that when we replace the negative samples in the test data with the ending generated by the model, the ending generated by the Seq2seq model are still treated as negative samples by the discriminator (N-score has no change), and the GPT[3] pre-trained model we use can mostly confuse the discriminator (N-score is close to 0.5). After replacing positive samples in the test data with the ending generated by the model, according to the P-score value, the discriminator tend to classify the ending generated by our model into positive

[2] https://github.com/ChemJeff/StoryCloze_ROCStories

[3] https://github.com/huggingface/pytorch-pretrained-BERT/blob/master/examples

samples, and the ending generated by Seq2seq are classified as negative samples. To make the results more objective, we use a robust discriminator that is also pre-trained by GPT to evaluate our model. As the result illustrated in Table 3, after replacing the negative samples, our model has reduced the correct rate of the robust discriminator from 0.87 to 0.53.

VI. Conclusion

We propose a new task (from quick-draw to story) as per our project needs. Combining the Google Quick Draw dataset with ROCStories corpus, we create training and test sets for new tasks and propose a Multi-task Transformer Network to meet our needs. Limited by the training data, we split the task into two subtasks: 1) generate a description by quick-draw. 2) generate a story based on the description. Compared with traditional RNN models, the transformer's structure can assist us in generating more fluid sentences. Combining related classification tasks can make our sentences focus on the object of quick-draw more accurately. In the story generation phase, the result of fine-tuning on the pre-trained model is more narrative than the traditional text-generating model.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[2] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank, "Automatic description generation from images: A survey of models, datasets, and evaluation measures," *Journal of Artificial Intelligence Research*, vol. 55, pp. 409–442, 2016.

[3] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5659–5667.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[5] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the ninth workshop on statistical machine translation*, 2014, pp. 376–380.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] A. Fan, M. Lewis, and Y. Dauphin, "Strategies for structuring story generation," *arXiv preprint arXiv:1902.01109*, 2019.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] T.-H. K. Huang, F. Ferraro, N. Mostafazadeh, I. Misra, A. Agrawal, J. Devlin, R. Girshick, X. He, P. Kohli, D. Batra *et al.*, "Visual storytelling," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1233–1239.

[10] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[12] B. Li, S. Lee-Urban, G. Johnston, and M. Riedl, "Story generation with crowdsourced plot graphs," in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[13] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004.

[14] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 375–383.

[15] L. J. Martin, P. Ammanabrolu, X. Wang, W. Hancock, S. Singh, B. Harrison, and M. O. Riedl, "Event representations for automated story generation with deep neural nets," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, "A corpus and evaluation framework for deeper understanding of commonsense stories," *arXiv preprint arXiv:1604.01696*, 2016.

[17] N. Mostafazadeh, M. Roth, A. Louis, N. Chambers, and J. Allen, "Lsdsem 2017 shared task: The story cloze test," in *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, 2017, pp. 46–51.

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "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, 2002, pp. 311–318.

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[20] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

[21] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7008–7024.

[22] M. O. Riedl and R. M. Young, "Narrative planning: Balancing plot and character," *Journal of Artificial Intelligence Research*, vol. 39, pp. 217–268, 2010.

[23] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.

[24] R. Swanson and A. S. Gordon, "Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 2, no. 3, p. 16, 2012.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[26] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[27] 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*, 2015, pp. 3156–3164.

[28] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[29] L. Yao, N. Peng, W. Ralph, K. Knight, D. Zhao, and R. Yan, "Plan-and-write: Towards better automatic storytelling," *arXiv preprint arXiv:1811.05701*, 2018.

[30] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[31] W. Zhao, B. Wang, J. Ye, M. Yang, Z. Zhao, R. Luo, and Y. Qiao, "A multi-task learning approach for image captioning." in *IJCAI*, 2018, pp. 1205–1211.