Generating Reasonable and Diversified Story Ending Using Sequence to Sequence Model with Adversarial Training

Zhongyang Li, Xiao Ding and Ting Liu*
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{zyli,xding,tliu}@ir.hit.edu.cn

Abstract

Story generation is a challenging problem in artificial intelligence (AI) and has received a lot of interests in the natural language processing (NLP) community. Most previous work tried to solve this problem using *Sequence to Sequence* (Seq2Seq) model trained with *Maximum Likelihood Estimation* (MLE). However, the pure MLE training objective much limits the power of Seq2Seq model in generating high-quality stories. In this paper, we propose using adversarial training augmented Seq2Seq model to generate reasonable and diversified story endings given a story context. Our model includes a generator that defines the policy of generating a story ending, and a discriminator that labels story endings as human-generated or machine-generated. Carefully designed human and automatic evaluation metrics demonstrate that our adversarial training augmented Seq2Seq model can generate more reasonable and diversified story endings compared to purely MLE-trained Seq2Seq model. Moreover, our model achieves better performance on the task of Story Cloze Test with an accuracy of 62.6% compared with state-of-the-art baseline methods.

1 Introduction

The task of story generation was first proposed in the field of artificial intelligence (AI) (Schank and Abelson, 1977). Recently, benefiting from deep learning technology, story generation again received increasing interests in the natural language processing (NLP) community. In this paper, we propose using adversarial training augmented Seq2Seq model to generate story ending given a story context. For example, given the story context "Sara had lost her cat. She was so sad! She put up signs all over the neighborhood. Then a wonderful thing happened.", our goal is to generate a possibly reasonable story ending "Somebody found her cat".

Much previous work in story generation or prediction focused on learning statistical models of event sequences from large-scale text corpora. Chambers and Jurafsky (2008) proposed unsupervised induction of *narrative event chains* from raw newswire texts, with *narrative cloze* as the evaluation metric. However, they utilized a very impoverished representation of events as the form of (*event, dependency*). To overcome the drawback of this event representation, Pichotta and Mooney (2014) presented a script learning approach that employed events with multiple arguments. Pichotta and Mooney (2016a) showed that the LSTM-based event sequence model outperformed previous co-occurrence-based methods for event prediction. However, this line of work build their models based on discrete verbs and tokens, which is far from being a complete sentence or a story.

There have been a number of recent work for story generation focusing on complete sentences. Kiros et al. (2015) described an approach for unsupervised learning of a generic, distributed sentence representations called *skip-thought* vectors. Such vectors can be used to predict neighboring sentences in the context. Pichotta and Mooney (2016b) used Seq2Seq framework directly operating on raw tokens to predict sentences, finding it is roughly comparable with systems operating on structured verb-argument

*Corresponding author: tliu@ir.hit.edu.cn

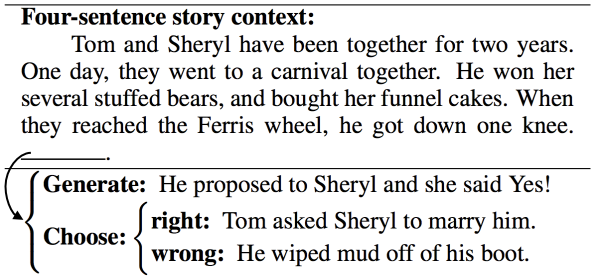

Figure 1: The task of generating or choosing the right story ending.

events in terms of predicting missing events in documents. Hu et al. (2017) also developed an end-to-end model for future subevent prediction. However, they use large-scale news corpus as training data, which is quite noisy and far from being reasonable stories. Moreover, traditional Seq2Seq model are usually trained purely by *Maximum Likelihood Estimation* (MLE). This makes sense in some tasks, such as, machine translation, where the gold-standard target sequence is unique, and the goal is to produce a target sequence as much like the gold-standard one as possible. However, this training objective is not enough for the task of story ending generation. Because this task essentially has no gold-standard answers, and any reasonable story ending can be the right one. On the other hand, purely MLE trained Seq2Seq model tends to generate frequent words or phrases in the test stage, which is a well known intractable obstacle. Hence, it is crucial to explore new ways to enhance Seq2Seq model in the task of story ending generation.

As shown in Figure 1, given a four-sentence story context, our target is to generate a possibly reasonable story ending. To this end, Story ending generation is modeled as a sequence to sequence generation process. In order to understand the context better and generate more reasonable and diversified story endings, we adopt the idea of adversarial training from *Generative Adversarial Nets* (Goodfellow et al., 2014) in recent image generation advances, which includes a generator that defines the probability of generating a story ending, and a discriminator that labels story endings as human-generated or machine-generated. In adversarial training augmented Seq2Seq model, the generator is encouraged to generate story endings that are indistinguishable from human generated story endings, and the discriminator gives a score of judging whether the current story ending is generated by the human or the machine, which can be used as a reward for the generator. Then the generator is trained to maximize the expected reward of the generated story ending using REINFORCE algorithm (Williams, 1992).

We conducted extensive experiments on the ROCStories corpus (Mostafazadeh et al., 2016). Carefully designed human evaluation demonstrates that adversarial training augmented Seq2Seq model can generate logically better story endings than conventional Seq2Seq model. Automatic evaluation metrics also suggest that adversarial training can improve the diversity of the generated story endings. Furthermore, we evaluate the effectiveness of our approach on the task of Story Cloze Test, which requires to choose the right story ending from two candidates given a story context. Our model chooses the right ending based on average word embedding similarities between our generated ending and the two candidates, and achieves the best performance on the task of Story Cloze Test compared to state-of-the-art baseline methods.

2 Sequence to Sequence Learning with Adversarial Training

As shown in Figure 2, our adversarial training augmented Seq2Seq model consists of three components. First, the generator $\mathcal{G}$ defines the policy that generates the story ending Y given the story context X. Second, the discriminator $\mathcal{D}$ is a binary classifier that takes as input a complete story $\{X, Y\}$ and outputs a label indicating whether the input is generated by human-beings or machines. The third component is an adversarial training process between the former two components.

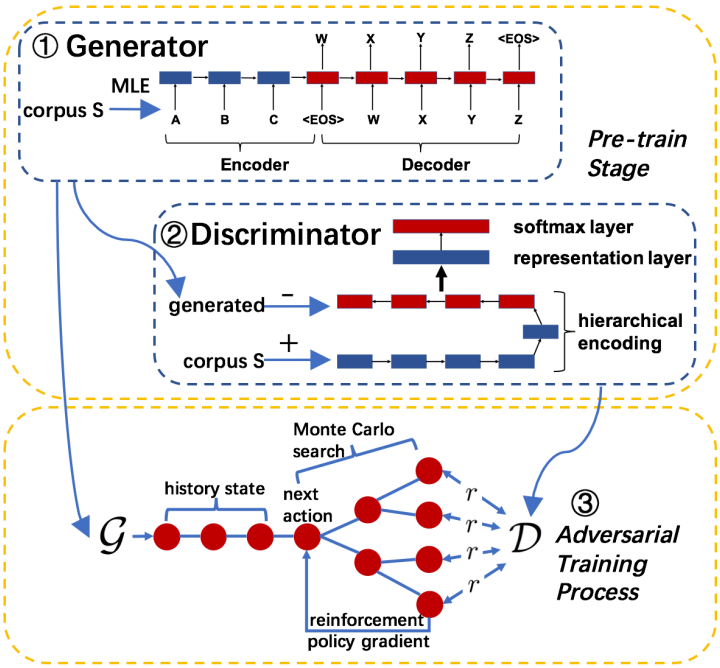

Figure 2: Framework of adversarial training augmented Seq2Seq model.

2.1 Sequence to Sequence Model as the Generator

In our task, the input is a four-sentence story context, and the output is a reasonable story ending. Hence, it is natural to use Seq2Seq model (Sutskever et al., 2014; Cho et al., 2014) as the generator (see ① in Figure 2). In Seq2Seq model, the story context is mapped to a vector representation using a recurrent neural network (Elman, 1990), and then the model computes the probability of generating each token in the target sequence (story ending) using a softmax function.

In recent work on Seq2Seq model, long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014) have shown their power on a wide variety of NLP tasks. Here, we use LSTM as the computation unit in our Seq2Seq generator, and sigmoid is used as the nonlinear activation function. Given a sequence of inputs $X = \{x_1, x_2, ..., x_N\}$, the generator defines a distribution over outputs and sequentially predicts tokens using a softmax function:

$$\begin{aligned} p(Y|X) &= \prod_{t=1}^{T} p(y_t|x_1, x_2, ..., x_N, y_1, y_2, ..., y_{t-1}) \\ &= \prod_{t=1}^{T} \frac{exp(f(h_{t-1}, e_{y_t}))}{\sum_{y'} exp(f(h_{t-1}, e_{y'}))} \end{aligned} \tag{1}$$

where $f(h_{t-1}, e_{y_t})$ denotes the activation function between h_{t-1} and e_{y_t} , and h_{t-1} is the representation output from the LSTM at time $t-1$. Each sentence concludes with a special end-of-sentence symbol *EOS*. Commonly, input and output use different LSTMs with separate compositional parameters to capture different compositional patterns.

In the decoding procedure, the algorithm terminates when an *EOS* token is predicted. Beam search is adopted for next word prediction, the embedding of which is then combined with preceding output for next step token prediction.

2.2 Discriminator

In the idea of adversarial training, we want to produce more natural, diversified and logically reasonable story endings. To this end, we introduce a discriminator (see ② in Figure 2) that judges whether the input is generated by human-beings or machines, and transfers this signal to the Seq2Seq generator, to help adjust its parameters for generating our expected story endings.

The discriminator $\mathcal{D}$ is a binary classifier. We use a hierarchical sequence encoder (Li et al., 2015) to map a complete story $\{X, Y\}$ into a vector representation, which is then fed to a 2-class softmax function, returning the probability of the story being a machine-generated story ending (denoted by $\mathcal{R}_-(\{X, Y\})$) or a human-generated one (denoted by $\mathcal{R}_+(\{X, Y\})$).

2.3 Adversarial Training Process

In the adversarial training augmented Seq2Seq model, the generator is encouraged to generate story endings that are indistinguishable from human generated story endings. We use *policy gradient* methods to achieve such goal (see ③ in Figure 2). The score $\mathcal{R}_+(\{X, Y\})$ of the current generated story ending is assigned by the discriminator. And then it is used as a reward for the generator. At last, the generator (we use θ to denote the parameters of the generator) is trained to maximize the expected reward of generated story ending using the REINFORCE algorithm (Williams, 1992):

$$J(\theta) = \mathbb{E}_{Y \sim p(Y|X)}(\mathcal{R}_+(\{X, Y\})|\theta). \tag{2}$$

We consider the story ending generation procedure as a sequential decision making process. Given the input story context X, the generator generates a story ending Y by sampling from the policy. The concatenation of the generated ending Y and the input X is fed to the discriminator.

Following Yu et al. (2016), we use reward for every intermediate token to update the generator. In our model, Monte Carlo (MC) search is used to assign the reward score for every intermediate token. Given a partially decoded sentence S, the model keeps sampling tokens from the generator until the decoding finishes. Such a process is repeated M times and the M generated sequences will share a common prefix S. These M sequences are fed to the discriminator, the average score of which is used as a reward for S.

For each machine-generated story ending Y, the discriminator gives a classification score $\mathcal{R}(X, Y)$. The gradient of eq. (2) is approximated using the likelihood ratio trick (Glynn, 1990; Williams, 1992), which is used to update the generator:

$$\begin{aligned} \nabla J(\theta) \approx & [\mathcal{R}_+(\{X, Y\}) - b(\{X, Y\})] \\ & \nabla \sum_t \log p(y_t|X, y_{1:t-1}), \end{aligned} \tag{3}$$

where $b(\{X, Y\})$ denotes the baseline value to reduce the variance of the estimate while keeping it unbiased. The discriminator is simultaneously updated with the human generated story ending including story context $\{X, Y\}$ as a positive example, and the machine-generated ending along with story context $\{X, \hat{Y}\}$ as a negative example.

To train the Seq2Seq model in adversarial manner, we need to first pre-train a Seq2Seq model on training corpus $\mathcal{S}$ with MLE. Then pre-train the discriminator use the instance $\{X, Y\}$ in training corpus $\mathcal{S}$ as positive examples, and generated ending along with story context $\{X, \hat{Y}\}$ as negative examples. Subsequently, the generator and discriminator are trained alternatively. During this process, discriminator $\mathcal{D}$ is first trained for *d-steps* with a similar way like pre-train stage. Then generator $\mathcal{G}$ is trained for *g-steps* using the REINFORCE algorithm. At last, we get the adversarially-trained generator $\mathcal{G}$.

3 Experiments

The performance of our adversarially-trained Seq2Seq model is compared with state-of-the-art baselines by evaluating the quality of generated sequences on two tasks: story ending generation and Story Cloze Test (choosing the right story ending from two candidates).

3.1 Dataset

The dataset used in this paper is ROCStories corpus, which is released by (Mostafazadeh et al., 2016). This corpus is a collection of five-sentence commonsense stories by crowd sourcing. Each story has the following major characteristics: (1) is realistic and non-fictional; (2) has a clear beginning and ending where something happens in between; (3) does not include anything irrelevant to the core story. These

	Training	Development	Test
#stories	98,161	1,871	1,871
#words	4,887,555	107,336	107,402

Table 1: Statistics of datasets.

stories are full of stereotypical causal and temporal relations between events, making them a great resource for commonsense reasoning and script knowledge learning. A two-step quality control step makes sure that there are no vague or boundary cases in the test dataset, making human performance of 100% accuracy possible.

The dataset is split into training, development and test datasets. Each instance in the training dataset is a five-sentence story. But in the development and test datasets, each instance is a four-sentence story and two candidate endings (one is the right ending and the other is the wrong ending). Specifically, the wrong endings are purposely designed to fit to the story context but logically wrong. Hence, sampling negative wrong endings from other stories is unreasonable. Detailed statistics of training, development and test datasets are shown in Table 1. All methods are evaluated on the test dataset, and only the development dataset could be used for tuning purposes.

3.2 Evaluation Metrics

For the task of Story Cloze Test, we use accuracy to evaluate the effectiveness of our approach. For the task of story ending generation, we ask human annotators to evaluate the performance of our approach, as it is difficult to find a generally accepted automatic metric for this task. Following Shang et al. (2015), we carefully designed the human evaluation procedure to evaluate the ability of our proposed model on generating story endings. We randomly pick 100 story endings generated by the baseline method and our approach on the test dataset respectively, and distribute them to three annotators. The annotators read the story context, and judge whether the generated story ending is appropriate and reasonable according to the story context. Four levels are assigned to each ending with scores from 0 to 3:

- **Bad (0)**: The ending doesn't make sense and unrelated to the story context.
- **Relevant (+1)**: The ending is partially related to the story context.
- **Good (+2)**: The ending is highly related to the story context.
- **Perfect (+3)**: The ending is high-quality, context-related and logically correct to the story context.

We give the following three aspects judgement criteria for the annotators: (a) **Grammar and Fluency:** Endings should be natural language and free of fluency and grammar errors. (b) **Context Relevance:** Person names, pronouns and phrases in the endings should be relevant to the story context. (c) **Logic Consistency:** Endings should be logically consistent with the story context.

Furthermore, we ask the annotators to directly compare the story endings that generated by the baseline method and our approach, and choose the better one.

We did not use perplexity or BLEU (Papineni et al., 2002) as evaluation metric, as neither of them is likely to be an effective evaluation metric in our scenario. Because our proposed model is designed to steer away from the standard Seq2Seq model, in order to generate more reasonable and diversified story endings. Following Li et al. (2016), we report the degree of diversity by calculating the number of distinct unigrams and bigrams in generated story endings. In order to avoid favoring long sentences, the value is scaled by total number of generated tokens.

3.3 Baselines and Proposed Models

For the evulation of story ending generation quality, the compared models are listed below:

- **Seq2Seq-MLE**: Seq2Seq model purely trained with MLE.
- **Seq2Seq-Adversarial**: Seq2Seq model pre-traind with MLE and then augmented with adversarial training.

	Models	
	Seq2Seq-MLE	**Seq2Seq-Adversarial**
Bad (0)	26.7%	17.7%
Relevant (+1)	23.3%	21.0%
Good (+2)	26.3%	27.0%
Perfect (+3)	23.7%	34.3%
Mean Score	1.47	**1.78 (+21.1%)**
Agreement	0.339	0.344

Table 2: Human evaluation results of story ending generation. Mean score is the average evaluation scores over three annotators.

Both good	6.7%
Both bad	11.3%
Seq2Seq-MLE is better	33.0%
Seq2Seq-Adversarial is better	**49.0**%
Agreement	0.443

Table 3: Pairwise model comparison of story ending generation.

We also evaluate the generated endings in the Story Cloze Test task, compare with the baseline methods used in (Mostafazadeh et al., 2016).

- **Word2Vec**: Choose the ending with closer average word2vec (Mikolov et al., 2013) to the average word2vec of four-sentence context.
- **Skip-thoughts**: (Kiros et al., 2015)'s Sentence2Vec encoder which models the semantic space of sentences, according to which we can choose the ending having a closer embedding to the four-sentence context.
- **Deep Structured Semantic Model (DSSM)**: MSR Sentence2Vector model (Huang et al., 2013), according to which we can choose the ending that has a closer embedding to the context.
- **Conditional Generative Adversarial Networks (CGAN)**: The Conditional GAN model proposed in (Wang et al., 2017), in which the discriminator is used to choose the correct story ending.

We choose the right story ending based on average embedding similarities between endings generated by **Seq2Seq-MLE** and **Seq2Seq-Adversarial** models, and the two candidates.

3.4 Results and Analysis

Human evaluation results of story ending generation are shown in Table 2 and Table 3. From Table 2, we find that Seq2Seq-Adversarial achieves better results than Seq2Seq-MLE model. Seq2Seq-Adversarial achieves a mean score of 1.78, which is 21.1% improvement over Seq2Seq-MLE (1.47). For the **Perfect** and **Good** levels, Seq2Seq-Adversarial outperforms Seq2Seq-MLE. While for the **Relevant** and **Bad** levels, Seq2Seq-Adversarial gets smaller percentage scores. The annotation agreements for Seq2Seq-MLE and Seq2Seq-Adversarial are evaluated by Fleiss' kappa (Fleiss, 1971), as a statistical measure of inter-rater consistency, which are 0.339 and 0.344, respectively. Considering the complex judgement criteria, they can demonstrate high agreement between three annotators (Shang et al., 2015).

As shown in Table 3, we directly compare the performance of Seq2Seq-MLE and Seq2Seq-Adversarial model. There are 49% story endings generated from Seq2Seq-Adversarial model better than that from Seq2Seq-MLE model. In turns, Seq2Seq-MLE model can only achieve better performance than

	UnigramDiv	**BigramDiv**
Seq2Seq-MLE	0.038	0.104
Seq2Seq-Adversarial	0.064 (+68.4%)	0.236 (+126.9%)

Table 4: Diversity evaluation of the generated endings on test dataset. **UnigramDiv** and **BigramDiv** are respectively the number of distinct unigrams and bigrams divided by total number of generated words.

Methods	Accuracy
Word2Vec (Mikolov et al., 2013)	53.9%
Skip-thoughts (Kiros et al., 2015)	55.2%
DSSM (Huang et al., 2013)	58.5%
Seq2Seq-MLE (Cho et al., 2014)	58.6%
CGAN (Wang et al., 2017)	60.9%
Seq2Seq-Adversarial (ours)	**62.6%**

Table 5: Experimental results of Story Cloze Test on test dataset.

Seq2Seq-Adversarial model on 33% story endings. And we get a higher Fleiss' kappa value for pairwise model comparison, which is up to 0.443.

Diversity evaluation results are shown in Table 4. We find that by integrating adversarial training, Seq2Seq model can generate more diversified story endings. Seq2Seq-Adversarial model achieves 68.4% higher unigram diversity and 126.9% higher bigram diversity score than Seq2Seq-MLE model respectively.

The main reasons for these results are as follows. First, MLE tends to generate constant phrases regardless of the story context. Especially when it reads an unfamiliar story context as input, it will generate some frequent phrases appearing in the training dataset, but unrelated to the story context. Second, adversarial training always tries to understand the whole semantics of story context and generates endings as human-beings. Hence, is will generate some context-related or even semantic and logically related endings. Third, adversarial training is dependent on MLE pre-trained Seq2Seq model. Hence, it inherits the advantages of MLE training. During the adversarial training process, it will update its parameters under another criteria (the reward scores given by the discriminator). Hence, adversarial training augmented Seq2Seq model is indeed an ensemble of two different training objectives. MLE training guarantees fluency of the generated endings, and adversarial training adds diversity to them by exploring more words in MC search process.

Results of Story Cloze Test on the ROCStories test dataset are shown in Table 5. The best baseline model is CGAN and it achieves an accuracy of 60.9%. This indicates the difficulty of this task. Seq2Seq-MLE (accuracy: 58.6%) achieves comparable result with the DSSM model (accuracy: 58.5%), and Seq2Seq-Adversarial gets the highest accuracy of 62.6% on the task of Story Cloze Test.

Indeed, all baseline models cannot generate story endings. They either try to directly compare the similarity between story context and two candidate endings or try to map the story context into a hidden vector representation, and then choose a similar ending. While our model try to address the problem of Story Cloze Test from a generation perspective. We first generate real story endings, and then choose the right ending based on the average embedding similarities between them and the candidates. This is easier to understand for readers. Because human readers can read both the generated endings and the candidates, and understand why we can choose the right ending.

Word2Vec method tends to choose the candidate ending whose average word embedding is more similar to the average word embedding of four-sentence context. Comparing with it, our model achieves a huge boost of accuracy (from 53.9% to 62.6%). This is mainly because our model can understand the global semantic of the story context, other than the local meaning of words. Our model achieves better performance than DSSM model, mainly because DSSM model needs to sample negative examples from other stories in its training process, but there are no negative examples in the training dataset. Our model also outperforms the CGAN model, which doesn't generate real story ending but use the discriminator to choose the correct one. This demonstrates the effectiveness of our model, which solve the problem based on real genrated endings.

3.5 Case Study

Three typical cases are shown in Table 6. In case one, both Seq2Seq-MLE and Seq2Seq-Adversarial models generate perfect story endings, that are fluent, grammatical and logically correct. In case two, Seq2Seq-MLE model generate relatively poor ending, which is not relevant to the story context. We argue that in the MLE-training process, it cannot see a similar story context in the training dataset, so

Case one	**Context**	Eric was overly excited for lunch today.The cafeteria was serving his absolute favorite meal.He loved the school's pizza and french fries.He sprinted to the lunch line but it was too late.
	Right	Eric had to settle for cold peanut butter and jelly for lunch.
	Wrong	Eric got a second plate of pizza and french fries.
	Seq2Seq-MLE	eric had to go to the store and buy a pizza instead.
	Seq2Seq-Adversarial	eric decided to go to the store for lunch instead.
Case two	**Context**	Aya wanted to paint a picture.She bought canvas and paints.Then she sat down by a window for inspiration.She began to paint an image of the landscape.
	Right	Aya became famous for her landscape pictures.
	Wrong	Aya put the finishing strokes on a picture of a skyscraper.
	Seq2Seq-MLE	aya was thrilled with her purchase.
	Seq2Seq-Adversarial	by the end of the day she had a picture.
Case three	**Context**	Ivy was scared to go to summer camp.But she steeled herself and got on the bus.When she got there, she went to talk to the other campers.Soon she had made a few new friends.
	Right	Ivy ended up loving summer camp.
	Wrong	But she wanted to go home.
	Seq2Seq-MLE	ivy had a great time with her friends.
	Seq2Seq-Adversarial	ivy decided to go to church instead.

Table 6: Example story endings generated by our models, along with the original right and wrong story endings in the test dataset.

it tends to generate some frequent phrases. While Seq2Seq-Adversarial model generates a perfect story ending, which really amazed us: it is even better than the original right story ending. Moreover, this ending is far from being similar to the gold-standard right ending. But it fits to the story context perfectly. Hence, we believe that Seq2Seq-Adversarial model can generate more diversiform knowledge for this world. In case three, Seq2Seq-MLE generates a better story ending than Seq2Seq-Adversarial. This is mainly because in the adversarial training process, the MC search sampled the *go to church* phrase and it got a high reward score from the discriminator. This can be improved by carefully controlling the MC search process and filtering the meaningless words or phrases.

4 Related Work

4.1 Script Learning

The use of scripts in AI dates back to the 1970s (Minksy, 1975; Schank and Abelson, 1977; Mooney and DeJong, 1985). In this conception, *scripts* are composed of complex events without probabilistic semantics. In recent years, a growing body of research has investigated learning probabilistic co-occurrence-based models with simpler events. Chambers and Jurafsky (2008) proposed unsupervised induction of *narrative event chains* from raw newswire text, with *narrative cloze* as the evaluation metric, and pioneered the recent line of work on statistical script learning (Jans et al., 2012; Pichotta and Mooney, 2014; Pichotta and Mooney, 2016b; Granroth-Wilding and Clark, 2016). However, they utilized a very impoverished representation of events as the form of (*event, dependency*). To overcome the drawback of this event representation, Pichotta and Mooney (2014) presented a script learning approach that employed events with multiple arguments.

There have been a number of recent neural models for script learning. Pichotta and Mooney (2016a) showed that the LSTM-based event sequence model outperformed previous co-occurrence-based methods for event prediction. Pichotta and Mooney (2016b) used a Seq2Seq model directly operating on raw tokens to predict sentences, finding it is roughly comparable with systems operating on structured verb-argument events. Granroth-Wilding and Clark (2016) described a feedforward neural network which composed verbs and arguments into low-dimensional vectors, evaluating on a multiple-choice version of the Narrative Cloze task. However, most of this line of work build their models based on discrete verbs and tokens, which is far from being a complete sentence or story segment as used in this paper.

A line of works has studied the problem of Story Cloze Test on ROCStories corpus (Mostafazadeh et al., 2016). Chaturvedi et al. (2017) explored three distinct semantic aspects including sequence of

events, emotional trajectory and plot consistency, and used a hidden variable coherence model to join these aspects together. Lin et al. (2017) adopted a similar method, and also exploited heterogeneous knowledge for this task. Wang et al. (2017) applied adversarial networks on this task, which is most similar to our work. However, all these studies put their focuses on choosing the correct story ending through discriminative approaches. We mainly aim to generate reasonable and diversified story endings.

4.2 Sequence to Sequence Learning

Sequence to sequence learning (Seq2Seq) aims to directly model the conditional probability $p(Y|X)$ of mapping an input sequence $X = \{x_1, ..., x_N\}$, into an output sequence $Y = \{y_1, ..., y_T\}$. It accomplishes such goal through the encoder-decoder framework proposed by (Sutskever et al., 2014) and (Cho et al., 2014). The encoder computes a representation s for each input sequence. Based on the input representation, the decoder generates an output sequence one word at a time.

A natural model for sequential data is the recurrent neural network (RNN) (Elman, 1990), which is used by most of the recent Seq2Seq work. These work, however, differ in terms of: (a) architecture - from unidirectional, to bidirectional, and deep multi-layer RNNs, (b) RNN type - which is long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) or gated recurrent unit (GRU) (Cho et al., 2014).

Another important difference among Seq2Seq work lies in what constitutes the input representations. The early Seq2Seq work (Sutskever et al., 2014; Cho et al., 2014) used only the last encoder state to initialize the decoder state. While, Bahdanau et al. (2015) proposed an attention mechanism, a way to provide Seq2Seq models with a random access memory, to handle long input sequences. Recent work such as (Xu et al., 2015; Yin et al., 2017; Ayana et al., 2017) has found that it was crucial to empower Seq2Seq models with the attention mechanism. Despite its success, many issues emerge due to its over-simplified training objective. In this paper, we propose training Seq2Seq model with adversarial strategy.

4.3 Generative Adversarial Networks

The idea of generative adversarial nets is proposed by (Goodfellow et al., 2014), which has achieved great success in computer vision (Mirza and Osindero, 2014; Radford et al., 2015). Training is formalized as a game in which the generator is trained to generate outputs fooling the discriminator.

However, this idea has not achieved comparable success in the NLP field. This is mainly due to the fact that unlike in image generation, the discrete property of text generation makes the error computed by the discriminator hard to backpropagate to the generator. Some recent work try to address this issue: (Lamb et al., 2016) proposed providing the discriminator with intermediate hidden vectors of the generator, which makes the system differentiable and achieves promising results in language modeling and handwriting generation tasks. Yu et al. (2016) used policy gradient reinforcement learning to back-propagate the error from the discriminator, showing improvement in multiple generation tasks such as poem, speech language and music generation. Li et al. (2017) used a similar strategy to boost the dialogue generation quality, which achieved good experimental results.

Not limited to the task of sequence generation, Chen et al. (2016) applied the idea of adversarial training to sentiment analysis, and (Zhang et al., 2017) investigated the problem of domain adaptation based on adversarial networks. To our knowledge, this is the first paper to study adversarial training augmented Seq2Seq model on the task of generating story endings.

5 Conclusion

Story generation is a challenging problem in AI. In this paper, we explore new ways to generate story ending given a four-sentence story context. In order to generate high-quality story endings, we adopt the idea of adversarial training from *Generative Adversarial Nets* and propose using adversarial training augmented Seq2Seq model to generate reasonable and diversified story endings. Carefully designed human evaluation shows that the adversarial training augmented Seq2Seq model can generate higher quality story endings than purely MLE-trained Seq2Seq model, with 21.1% mean human annotation score improvement and 126.9% bigram diversity improvement. Furthermore, our model can choose the

right ending based on the generated story ending, and achieves the best performance. This verifies the potential of solving the Story Cloze Test problem from a generation perspective.

Acknowledgements

This work is supported by the National Key Basic Research Program of China via grant 2014CB340503 and the National Natural Science Foundation of China (NSFC) via grants 61472107 and 61702137. The authors would like to thank the anonymous reviewers for their insightful comments.

References

Ayana, Shi-Qi Shen, Yan-Kai Lin, Cun-Chao Tu, Yu Zhao, Zhi-Yuan Liu, Mao-Song Sun, et al. 2017. Recent advances on neural headline generation. *Journal of Computer Science and Technology*, 32(4):768–784.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR*.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL*, volume 94305, pages 789–797.

Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story comprehension for predicting what happens next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1604–1615, Copenhagen, Denmark, September. Association for Computational Linguistics.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2016. Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP*, pages 1724–1734.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Peter W Glynn. 1990. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*, pages 2672–2680.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *AAAI*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Linmei Hu, Juanzi Li, Liqiang Nie, Xiao-Li Li, and Chao Shao. 2017. What happens next? future subevent prediction using contextual hierarchical lstm. In *AAAI*, pages 3450–3456.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, pages 2333–2338. ACM.

Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *EACL*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*, pages 3294–3302.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *NIPS*, pages 4601–4609.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *ACL*, pages 1106–1115.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. *NAACL*, pages 110–119.

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *EMNLP*, pages 2147–2159.

Hongyu Lin, Le Sun, and Xianpei Han. 2017. Reasoning with heterogeneous knowledge for commonsense machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2022–2033, Copenhagen, Denmark, September. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Marvin Minksy. 1975. A framework for representing knowledge. *The psychology of computer vision*, 73:211–277.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *Computer Science*, pages 2672–2680.

Raymond Mooney and Gerald DeJong. 1985. Learning schemata for natural language processing. *Urbana*, 51:61801.

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. *NAACL*, pages 740–750.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Karl Pichotta and Raymond J Mooney. 2014. Statistical script learning with multi-argument events. In *EACL*, volume 14, pages 220–229.

Karl Pichotta and Raymond J Mooney. 2016a. Learning statistical scripts with lstm recurrent neural networks. In *AAAI*.

Karl Pichotta and Raymond J Mooney. 2016b. Using sentence-level lstm language models for script inference. *ACL*, pages 279–289.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Roger C Schank and Robert P Abelson. 1977. Scripts, plans, goals, and understanding: An inquiry into human knowledge structures (artificial intelligence series).

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *ACL*, pages 1577–1586.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Bingning Wang, Kang Liu, and Jun Zhao. 2017. Conditional generative adversarial networks for commonsense machine comprehension. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4123–4129.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, pages 77–81.

Jun Yin, Wayne Xin Zhao, and Xiao-Ming Li. 2017. Type-aware question answering over knowledge base with attention-based tree-structured neural networks. *Journal of Computer Science and Technology*, 32(4):805–813.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2016. Seqgan: sequence generative adversarial nets with policy gradient. *AAAI*.

Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2017. Aspect-augmented adversarial networks for domain adaptation. *TACL*.