Conditional Generative Adversarial Networks for Commonsense Machine Comprehension

Bingning Wang[1,2], **Kang Liu**[1], and **Jun Zhao**[1,2]
[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
{bingning.wang, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Recently proposed *Story Cloze Test* [Mostafazadeh *et al.*, 2016] is a commonsense machine comprehension application to deal with natural language understanding problem. This dataset contains a lot of story tests which require commonsense inference ability. Unfortunately, the training data is almost unsupervised where each context document followed with only one positive sentence that can be inferred from the context. However, in the testing period, we must make inference from two candidate sentences. To tackle this problem, we employ the generative adversarial networks (GANs) to generate fake sentence. We proposed a **C**onditional **GAN**s (CGANs) in which the generator is *conditioned* by the context. Our experiments show the advantage of the CGANs in discriminating sentence and achieve state-of-the-art results in commonsense story reading comprehension task compared with previous feature engineering and deep learning methods.

1 Introduction

Machine comprehension (MC) is one of the primary goals in Artificial Intelligence and Natural Language Processing (NLP). Commonly, MC requires two types of ability. **Retrieval:** The system needs to find answers from related documents or knowledge bases, and questions in this type of MC sometimes are factoid such as 'Who, Where, When' etc. Some MC datasets and evaluations have been proposed focusing on this type of task, for example, CNN/Daily Mail [Hermann *et al.*, 2015] and Children Book Test [Hill *et al.*, 2015] where one must fill a noun in a cloze style sentence based on the context; Or SQuAD [Rajpurkar *et al.*, 2016] and NewsQA [Trischler *et al.*, 2016] where the answer contains multiple words. **Inference:** the system needs to find clues in the documents and make inference based on them. This type of questions are sometimes non-factoid such as 'Why, How'; MCTest [Richardson *et al.*, 2013] is a typical dataset where more than half of the questions are multi-sentence-supported and we must make inference among multiple sentences; Story Cloze Test (SCT) [Mostafazadeh *et al.*, 2016] is a recently proposed MC task which consists of a lot of human-created stories, each story consists of 5 sentences that capture a variety of causal and temporal relations between everyday events and enables learning narrative structures across a range of events rather than a single domain or genre. The sentence in SCT are highly recapitulative, an example of SCT story is shown in Figure 1.

SCT contains a lot of entailments that can not be directly inferred from the text, for example, given '*The room is out of power*', one can not directly infer '*he light up a candle*' just from the text evidence, so the inference from implicit commonsense knowledge is necessary. Traditional feature engineering methods which utilize shallow linguistic features may not capture this type of patterns, a host of baselines based on shallow language understanding struggle to achieve a high score on this dataset [Mostafazadeh *et al.*, 2016]. Recent years deep-learning architectures have shown great advantage in representing the meaning of word and sentence, and achieved good results in many NLP inference tasks such as question answering [Hermann *et al.*, 2015; Seo *et al.*, 2016], recognizing textual entailments [Bowman *et al.*, 2016] and answer selection [Santos *et al.*, 2016]. However, the SCT training data only contains the positive examples that makes the standard discriminative neural networks based systems hard to apply.

To generate negative sentences that can make a discriminative classifier available, in this work we use the generative adversarial networks (GANs) [Goodfellow *et al.*, 2014], a generative framework under an adversarial process to generate the negative examples. In GANs two types of models are trained simultaneously: a generative model G to estimate the data distribution from random noise and generate a fake sample, and a discriminative model D to discriminate the real sample from the fake one. GANs corresponds to a minimax two-player game where there exists a unique solution that G recovers the data distribution and D equals to 1/2 everywhere, and this training process results in an optimal D that could discriminate the real target from the wrong one. In our task the generated sentence is not independent but conditioned on the contextual 4 sentences, so we modify the generator that the hidden variables are not only drawn from random noise but also the representation of the context (so we name our model *conditional* GANs). The discriminator is made up of three parts: a long-short-term-memory recurrent neural networks (LSTM-RNN) model [Hochreiter and Schmidhuber,

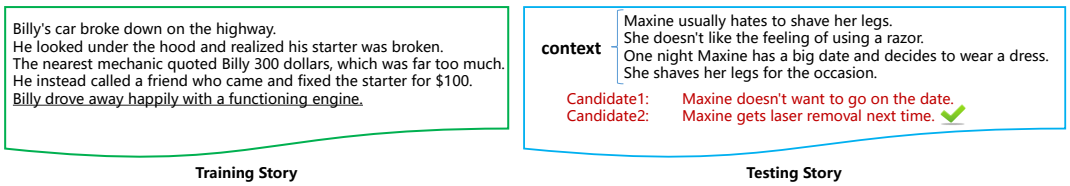

Figure 1: An example of Story Cloze Test. In test period, based on the context 4 sentences we need to choose from two candidate sentences which one can be inferred, however, during training only the positive target (underlined) is provided.

1997] to represent the sentence; an attention-based LSTM-RNN model to represent the document; a bilinear model to calculate the context document and target sentence similarity. The objective of the discriminator is to make its estimated score of the real target sentence high while reducing the score of the fake sentence generated by G, and the generator tries to '*fool*' the discriminator to make the fake sentence it generated score high. G and D are trained in an alternating way which results in better and better behavior.

However, traditional GANs can not be applied to text: On the one hand, when the output of the generator is discrete, it is impossible to pass the gradient update from the discriminator to the generator. On the other hand, the success of GANs is dependent on the Nash equilibrium point of a non-convex game with continuous, high dimensional parameters, but this point is notoriously hard to find [Donahue *et al.*, 2016; Arjovsky and Bottou, 2017]. In this paper, to solve the first problem, we utilize the Gumbel-softmax [Jang *et al.*, 2016; Kusner and Hernández-Lobato, 2016] on generator to make the generated output continuous, however, different from the previous method that set the temperature value in Gumbel-softmax by rule of thumb, we take the value as a model parameter and fit it automatically. To solve the second problem, we pre-trained G with maximum likelihood estimation (MLE), and after a few epochs we add some noise to the discriminator to make the downstream generator more stable.

Our experimental results show the advantage of the CGANs compared with other deep-learning systems and achieves a new state-of-the-arts result in SCT. In addition, as our CGANs is unsupervised, we conduct a preliminary experiment trying to employ the external large unlabeled corpus to enhance the behavior of our CGANs. Although it does not perform so well but shine light for future work. We do some ablation experiments to highlight the advantage of CGANs and the attention mechanism. At last, we analyze the commonsense MC problem and show the difficulty to solve them using off-the-shelf systems.

2 Methodology

Generative Adversarial Networks [Goodfellow *et al.*, 2014] are a class of methods for learning generative models based on game theory. This type of architecture consists of two separate models: generator network $G(z;\boldsymbol{\theta}^G)$ and discriminator network $D(x;\boldsymbol{\theta}^D)$. The generator transforms a vector noise z into a fake data $\overline{x}$, the discriminator tries to minimize the probability of $\overline{x}$ and increase the probability of the real data x. The generator tries to increase the probability of the fake data $\overline{x}$. Training GANs is equal to finding a Nash equilibrium between the two non-cooperative player which is formulated as follows:

$$\begin{aligned}\min_G \max_D V(D,G) = &\mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \\ &+ \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]\end{aligned} \tag{1}$$

However, in practice Equation 1 may not provide sufficient gradient for G. So sometimes we train the generator to maximize $\log D(G(x))$ rather than minimize $\log(1-D(G(z)))$.

2.1 Discriminator Network

The discriminator network consists of two sub-networks: a sentence representation network to model sentence from input word embedding and a document representation network to model the document from sentence representation.

Sentence representation: In this paper we use recurrent neural networks to model the variable-length text. Instead of LSTMs, we adopt a LSTM variant called gated recurrent unit (GRU) [Cho *et al.*, 2014] as building block for RNN because it has shown advantages in many tasks and comparative less parameter [Jozefowicz *et al.*, 2015]. After recurrently processing the words sequence we use the last word hidden representation as the sentence representation.

Document representation: we use another GRU to process each sentence representation derived from previous step recurrently to get the context document representation. However, as in our commonsense MC application, the context document should be represented based on the target sentence. For instance, in the left example of Figure 1, the last sentence (underlined) has a noun phrase '*functioning engine*', so when building the context document representation we should focus on the 4th sentence that contains '*fixed the starter*'. In this paper we add the recently well-developed **attention** mechanism [Bahdanau *et al.*, 2014; Wang *et al.*, 2016c] into the document representation process. Concretely, when adding the sentence representation to the document-GRU hidden unit, we *gate* each sentence representation with respect to the attention from the target sentence as follows:

$$\begin{aligned}\alpha_t &= \sigma(\mathbf{r}_4^T \mathbf{W}_a \mathbf{r}_t) \\ \tilde{\mathbf{r}}_t &= \alpha_t \odot \mathbf{r}_t\end{aligned} \tag{2}$$

where $\mathbf{W}_a$ is the bilinear attention matrix and $t \in [0,1,2,3]$, $\mathbf{r}_t$ is the target sentence representation. We omit the superscript s (denote 'sentence') for simplicity. After the attention process, we take the gated sentence representation (i.e. $\tilde{\mathbf{r}}_t$) as input to another GRU which we call document GRU. We use the last document GRU hidden state $\mathbf{r}_3^d$ as document representation $\mathbf{r}^d$. Finally, we use a matrix $\mathbf{M}$ to transform the document representation into the sentence representation space and the document-entail-target probability can be denoted as

their dot sigmoid value:

$$SCORE = \sigma[(\mathbf{r}^d)^T \mathbf{M} \mathbf{r}_4^s] \tag{3}$$

The discriminator architecture is illustrated in Figure 2.

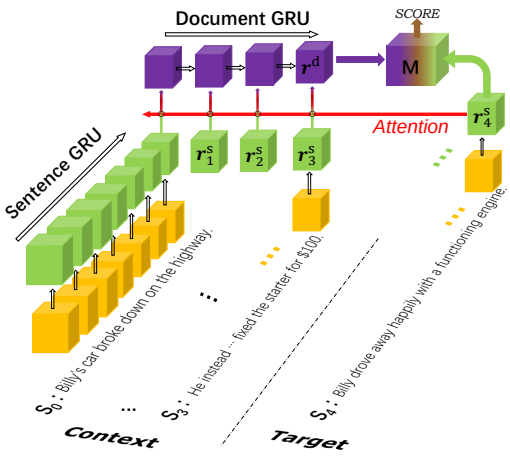

Figure 2: Discriminator Network. Yellow blocks denote word embedding, green and purple blocks stand for sentence and document GRU hidden states respectively.

2.2 Generator Network

The goal of our generator network is to generate a fake sentence and deceive the discriminator to take it as real target. In original GANs the input to the generator is only a random noise **z**, but in our application the generated fake sentence must in accordance with the context, so we fed context representation $\mathbf{r}^d$ as an additional input to G. Intuitively, the generator could be a standard decoder as follows:

$$\begin{aligned} \mathbf{z} &\sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \\ \mathbf{h}_i^d &= GRU(\mathbf{y}_{i-1}, \mathbf{h}_{i-1}; \mathbf{z}, \mathbf{r}^d) \\ \mathbf{y}_i &= \underset{j \in 1,\ldots,|V|}{\text{argmax}}\ softmax(\mathbf{D}_j^T \mathbf{W}_p \mathbf{h}_i^d) \end{aligned} \tag{4}$$

Where the **D** is word embedding lookup matrix with $|V|$ words, $\mathbf{W}_p$ is projection matrix that transform the generator hidden space to word embedding space. However, the argmax operation in Equation 4 makes the output discrete that the gradient of the discriminator could not be applied to the generator. Thus we use the Gumbel-softmax [Jang *et al.*, 2016] to replace the softmax+argmax operation to get the output word embedding. Concretely, we calculate the output word embedding **y** as follows:

$$\begin{aligned} \pi_j &= softmax(\mathbf{D}_j^T \mathbf{W}_p \mathbf{h}_i^d) \\ g_j &= -\log(-\log(u)), u \sim Uniform(0,1) \\ p_j &= softmax(\frac{\log(\pi_j) + g_j}{\tau}) \\ \mathbf{y} &= \sum_{j=1}^{|V|} p_j \mathbf{D}_j \end{aligned} \tag{5}$$

g_j is a sample from $Gumbel(0,1)$ distribution. τ is the temperature of the softmax that when it is set to zero, the output of Gumbel-Softmax is one-hot which is identical to the categorical distribution $p(\pi_i)$. And when $\tau \to \infty$, the distribution is just uniform and the output word embedding is the mean of all word embedding in the vocabulary. In this paper, we calculate τ as follow:

$$\tau = Relu(\mathbf{w}_\tau^T \mathbf{h}_i^d) + \varepsilon \tag{6}$$

Where $\mathbf{w}_\tau^T$ is a vector to calculate τ and ε is a small value to keep τ positive.In this way, the temperature of the model could be determined by itself.

In this paper, to make the training process more stable, we make four improvements:

- Instead of training the generator from random initiation, we pre-trained it by maximum the sentence likelihood which corresponds to neural machine translation the ground truth sentence is fed to the generator at each time step as a supervision. The context information is ignored during pre-training process and only the random vector is input to the generator.
- Adding small noise to the inputs of the discriminator to smooth the distribution of the generator output probability mass. As has been proposed in [Arjovsky and Bottou, 2017], the support of the data distribution and generator distribution are disjoint or lie on low dimensional manifolds, thus the gradient of the discriminator to the generator will be zero almost everywhere. Thus adding some noise makes their support intersecting[1].
- During training process, unlike MLE where the supervision is imposed on every step of the decoder, in CGANs only the score of the discriminator are provided. In order to train the generator more quickly, we add more supervision to the generator: we force the generated sentence to have more semantic similarity with the real target sentence. We do it by adding another objective to the generator in Equation 1: $similarity(s_4, \overline{s}) = 1 - cosine(r_{\overline{s}}^s, r_4^s)$ where $\overline{s}$ is the generated sentence.
- Sometimes training GANs is unstable which may cause the loss of G and D departure a lot, to monitor the training process of each part we calculate their loss and determine the number to update them: If the loss of G is much larger than D then we update G more often and vise versa.

The training process of CGANs is detailed in Algorithm 1.

3 Experiment

We use the off-the-shelf 100-dimensional word embeddings from word2vec website[2] and fix it during training. All weight and attention matrices are initiated by fixing their largest singular values to 1.0. We use Adadelta with $\rho = 0.999$ to update parameter. We use L_1 criteria with weight 1e-5 to regulate the parameter. All training process is implemented with batch size equals to 32. For the discriminator: we set the vocabulary size to 25000 after pre-processing. The sentence GRU hidden

[1]We do this optimization only after few epochs when the discriminator is somewhat '*perfect*'.

[2]https://code.google.com/archive/p/word2vec/

	Random	Frequency	N-gram-overlap	Gensim	Sentiment-Full	Sentiment-Last	Skip-thoughts	Narrative-Chains-AP	Narrative-Chains-Stories	DSSM	GRU	w/o CGAN&Attention	w/o Attention	w/o CGAN	CGAN
Validation Set	0.514	0.506	0.477	0.545	0.489	0.514	0.536	0.472	0.510	0.604	0.573	0.589	0.603	0.593	**0.625**
Test Set	0.513	0.520	0.494	0.539	0.492	0.522	0.552	0.478	0.494	0.585	0.561	0.580	0.595	0.578	**0.609**

Table 1: Accuracy of different methods in SCT.

Algorithm 1 Conditional Generative Adversarial Networks

Require:$Threashold$: # of iteration to add noise. k_d=k_g=1 .

Pre-training G by maximizing target sentence likelihood.
for number of training iterations **do**
  for number steps k_d **do**
    sample a context document $\{s_0, ..., s_3\}$ and the real target s_4, then use G to generate fake sentence $\overline{s}$
    if iteration $> Threashold$ **then** sample $\mathbf{z}_d$ from $\mathcal{N}(\mathbf{0}, \mathbf{1})$, add $\mathbf{z}_d$ to the embedding of s_4 or $\overline{s}$.
    Calculate $SCORE$ from Equation 3.
    $\mathcal{L}_D = -\log SCORE(s_4) - \log(1 - SCORE(\overline{s}))$
    $\triangleright$ Update the discriminator:
    $\nabla_{\theta_D} = \frac{d\mathcal{L}_D}{d\theta_D} \qquad \theta_D = \theta_D + \lambda \nabla_{\theta_D}$
  for number steps k_g **do**
    sample a context document $\{s_0, ..., s_3\}$ and get the generated sentence $\overline{s}$ from generator G
    $\mathcal{L}_G = -\log SCORE(\overline{s}) + similarity(s_4, \overline{s})$
    $\triangleright$ Update the generator:
    $\nabla_{\theta_G} = \frac{d\mathcal{L}_G}{d\theta_G} \qquad \theta_G = \theta_G + \lambda \nabla_{\theta_G}$
  $\triangleright$ Update k_g and k_d based on $\mathcal{L}_G$ and $\mathcal{L}_D$

state size is set to 128 and the document hidden state size is set to 150. For the generator: the decoder size is set to 256, and we use a transfer matrix to project the document representation (150d) into the decoder space. ε is set to 1.0E-20. If G has not yet generate word 'STOP' after 50 steps, then we stop it. The THRESHOLD was set to 0.2. For k_d and k_g, we truncate their max value to 20 (i.e. at most 20 samples are fed to G or D every training step).

3.1 Baselines

There are 10 baseline methods proposed in [Mostafazadeh *et al.*, 2016] (a) **Random**: random select a sentence from the two candidates. (b) **Frequency**: using TRIPS semantic parser to extract the target sentence main verb, and then select the one whose main verb that get more hits from search engine. (c) **N-gram Overlap**: choose the alternative which shares more n-grams with the context, the n-grams were calculated using Smoothed-BLEU [Lin and Och, 2004] and n is set to 4. (d) **GenSim: Average Word2Vec**: choose the candidate with its average word embedding closer to the context. (e) **Sentiment-Full**: Choose the hypothesis that matches the average sentiment of the context using the state-of-the-art sentiment analysis model [Manning *et al.*, 2014] which assigns a numerical value from 1 to 5 to a sentence. (f) **Sentiment-Last:** Choose the hypothesis that matches the sentiment of the last context sentence. (g) **Skip-thoughts Model**: This model uses Skip-thoughts Sentence2Vec embedding [Kiros *et al.*, 2015] which use a RNN encoder to encode the source sentence and two decoders to predict the previous and following sentence. (h) **Narrative Chains-AP**: Implements the standard approach to learning chains of narrative events based on Chambers and Jurafsky [2008] and choose the hypothesis whose co-referring entity has the highest average PMI score with the entitys chain in the context. (i) **Narrative Chains-Stories:** The same model as above but trained on Story Cloze Test. (j) **Deep Structured Semantic Model (DSSM)**: This model is trained to project the context and the fifth sentence into the same vector space [Huang *et al.*, 2013].

In addition to the above baselines, we also conduct some ablation experiments to evaluate the importance of the adversarial training and the improvement by applying attention mechanism. The model without CGANs is reduced to a single discriminator that no negative sentence is provided by the generator, so we randomly sample a sentence from the training dataset as the negative target and train the discriminator thereof. For the model without attention mechanism, the input to the document GRU is the original sentence representation $\mathbf{r}_t$ without the attention gate operation in Equation 2. The result is shown in Table 1.

4 Analysis

CGANs: As shown in Table 1, our proposed sentence-document GRU model is competitive compared with previous methods. However, when equipped with CGAN, the performance increases a lot and achieves a new state-of-the-arts result. When the discriminator is trained based on random negative samples (GRU, w/o CGANs) the performance drops a lot, we conjecture that the sampled negative target sentences have huge semantic distance with the real target so the boundary of the discriminator to discriminate the positive from negative is weak. However, the negative sentence in test set has a relatively close distance to the real one in the semantical space thus the blurred discriminator fails to make discrimination. The improvement of CGANs may attribute to the advantages of the generative network and the adversarial training process where we get more *challenging* negative sentences that make the discriminator more discriminative. In order to analysis the improvement of CGANs quantitatively, we start with a pre-trained sentence-document GRU model and evaluate the CGANs every 10000 samples, then we average the normalized Euclidean distance between the true and generated target sentence, the result is shown in Figure 3.

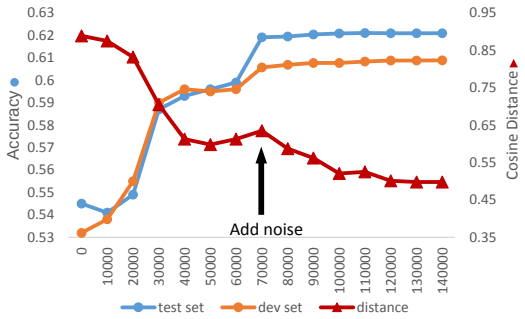

Figure 3: Performance with respect to the training process. X-axis is the training examples that had been fed to the discriminator. The red line is the normalized Euclidian distance between true target sentence and generated fake sentence.

At the beginning of CGANs training process, our model does not improve significantly because the generator does not learn very well to generate the competitive negative examples. With the training proceeds and the generator produces negative sentences with higher quality (i.e. Euclidean distance with the real target became smaller), the behavior of the discriminator improves accordingly. In addition, after few epochs, we add some noise to the discriminator to make the generated sentence and the real sentence have intersecting support and therefore benefit the performance.

In order to measure the improvement by introducing noise to the discriminator, we conduct further experiment: we train two separate CGANs on SCT, however, the second model is trained without any noise. We report the loss for both discriminator and generator, the result is shown in Figure 4.

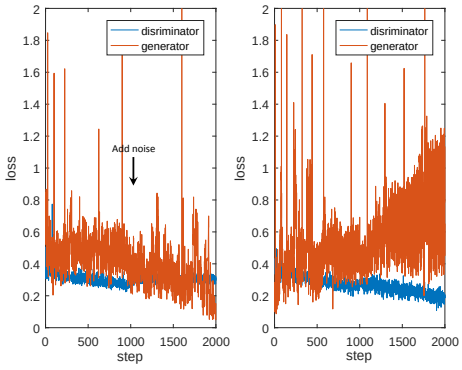

Figure 4: The training loss w.r.t. discriminator and generator. The right model is not fed with noise. We add noise to the from step 1000 to the end, one step equals to 8 batches.

We can see that after adding some noise to the discriminator, although it is impacted but the generator becomes more and more stable and optimal, thus it could generate more competitive sentences to make the discriminator more discriminative. If we do not add noise during training, after a few steps the discriminator is optimal and the generator is divergent. We suppose that the random noise could '*push*' the discriminator from one point where it may have stuck in, and thus its gradient could be properly transmitted to the generator.

Attention Based Sentence-Document GRU: In this work we build a document embedding model in a hierarchical manner in which the sentence and the document are represented separately. However, previous works usually employ a single RNN to model the whole document [Hermann *et al.*, 2015]. In order to evaluate the improvement by introducing the hierarchical structure, we build a single GRU network similar to the one proposed in Hermann et al. [2015] and show the result in Table 1 (GRU). The poor performance of a single GRU may attribute to the fact that sentences in SCT are not semantically continuous and sometimes there exists transition between consecutive sentences. Take the right example in Figure 1 for instance, the second sentence '*She doesnt like the feeling of using a razor*' and the third sentence '*One night Maxine has a big date and decides to wear a dress*' are not semantically continuous and there is no discourse relationship between them, a single GRU network is not good at modeling this linguistic phenomenon.

In addition, it can be shown in the Table 1 that the attention mechanism could benefit our model a lot, it has been proved in many previous works that the attention mechanism could enhance the semantic-inference ability of a neural model [Luong *et al.*, 2015]. We illustrate an example in Figure 5.

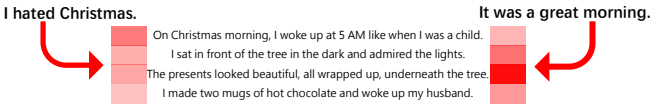

Figure 5: An example drawn from the validation set. The red rectangle denotes the attention weight. Deeper color means more attention.

It can be seen from the figure that when context document sentence is related to the candidate target sentence, this sentence will get more attention and weight more in the final document representation, and in SCT not all context information should be taken into account for inference, thus the *sentence distillation* process makes the subsequent inference easier.

Unsupervised pre-training is a promising ameliorates to our model. As our model is nearly unsupervised which only requires <context,target> pairs, so it is extensible to the large unsupervised data such as newspaper or wiki article where several coherent events are connected in the same document. In this work, we pre-train our CGANs in the New York Times (NYT) news article corpus[3]. We filter out some documents that are too short to process and get nearly 220,000 documents which contain about 8,000,000 sentences. Without loss of generality, we select four consecutive sentences in a document as context and the next sentence as the target. The experiment detail is same with SCT. In this paper, we design several setups on the external unlabeled data: **I**: training the CGANs on NYT and test it on SCT. **II**: pre-training the CGANs on small portion of NYT (nearly 20000 examples) then fine-tune it on SCT. **III**: pre-training the CGANs on entire NYT set and then fine-tuned it on SCT. The result is shown in Table 2.

[3]https://catalog.ldc.upenn.edu/LDC2008T19

	I	II	III
Dev	0.549	0.594	0.558
Test	0.538	0.588	0.574

Table 2: The result of CGANs with external unlabeled data.

We can find that the performance drops a lot when employing the external data, this is inconsistent with our expectations. We find that the average sentence length of NYT is 21.8 while the average sentence length of SCT is 7.5 and the pre-trained generator of CGAN is prone to generate long sentence. And the sentences in SCT are highly summary and abstractive, but in NYT the sentences are narrative and each sentence contains a lot of information that may be redundant to the topic or sub-sequent sentence. In addition, the events in NYT are limited to few topics such as '*kill, dead, explode*' etc., however the causal and temporal events in SCT span larger domain. This inherent divergence may cause the poor performance in our task.

The difficulty of commonsense MC: Commonsense is the basic ability to perceive, understand, and judge things. It is formed and developed during our daily experience and finally be expected by all people without the need for debate. Humans could achieve 100% accuracy in the SCT while state-of-the-arts models struggle to outperform 60%. Deep analyze the SCT data we found that many stories in commonsense MC are beyond lexical matching or even semantic inference. For example:

1) Morgan enjoyed long walks on the beach.
2) She and her boyfriend decided to go for a long walk.
3) After walking for over a mile, something happened.
4) Morgan decided to propose to her boyfriend.
⇒ Her boyfriend was upset he didn't propose first.

Inference like this is easy for our adults but really hard for a machine. The commonsense is innumerable and can not be covered by limited textual evidence, and other forms of knowledge such as ethical or scientific evidence is required to strengthen the commonsense inference ability of a system.

5 Related Work

Generated adversarial networks were proposed by Goodfellow [2014] as a generative model. Compared with other generative models such as variational auto-encoder [Kingma and Welling, 2013], GANs could generate higher quality images. Since then, many applications or improvement have been applied to GANs such as LAPGAN [Denton *et al.*, 2015] which generate images in a coarse-to-fine fashion by generating and upsampling in multiple steps; InfoGAN [Chen *et al.*, 2016b], an information-theoretic extension to the GANs that is able to learn disentangled representations. [Yu *et al.*, 2017] is the first work as far as we know to apply GANs on text, however, they train it by policy gradient which may take a long time to fit the model.

Gumbel-softmax is could make our gradient descent method applicable. it is actually a *re-parameterization* trick for a distribution that we can smoothly deform into the categorical distribution. Using the Gumbel-Max trick could provide an efficient way to draw samples from the categorical distribution [Jang *et al.*, 2016]. Traditional methods to deal with the discrete prediction are sometimes depend on REINFORCE algorithm [Williams, 1992; Yu *et al.*, 2017], however, as this algorithm relies on sampling and Monte Carlo search which decrease the training speed drastically. In addition, this algorithm sometimes requires a strong *baseline* to reduce the variance of the gradient but this baseline is hard to derive and sometimes have a negative impact on the training process.

Machine comprehension is a recently proposed natural language understanding task which follows the traditional QA. Traditional method mainly focuses on employing off-the-shelf NLP tools to extract features such as POS tags and then build a feature engineering system [Wang *et al.*, 2016a; 2016b]. Herman et al. [2015] proposed a large cloze style dataset CNN/Daily Mail dataset in which the target is to generate word in a statement slots given the context. Memory networks based models have been proposed to solve question answering problem [Weston *et al.*,] on bAbi [Weston *et al.*, 2015]. However, this dataset is synthesized and only contains a small vocabulary, so a rule-based system solves them nearly totally correct [Lee *et al.*, 2015]. SQuad [Rajpurkar *et al.*, 2016] and NewsQA [Trischler *et al.*, 2016] are recently released MC dataset in which only the question and document are given so we must predict the answer from the document. This type of dataset is sometimes based on formal text such as wiki or news articles and most of the questions are limited to syntactic variation or lexical variation. In this work we are focused on commonsense MC which evaluates systems deep semantic inference ability. The baseline models proposed in [Mostafazadeh *et al.*, 2016] containing not only feature engineering systems but also deep learning models, but the performance is still poor compared with human.

6 Conclusion

In this work, we propose a sentence-document GRU models with discriminative adversarial training. Our experimental result demonstrates the advantage of adversarial training and achieves a new state-of-the-art result in commonsense machine comprehension task. We also introduce the attention mechanism that could benefit our document representation. However, our model is still poor compared with human beings, we found that much inference in commonsense MC is too hard for a machine system to induce and deduce. Although in this work we fail to benefit from the external large unsupervised data, in the future we plan to introduce more consistent unlabeled texts such as novel or design a better mechanism that could employ the external knowledge and abstract *commonsense* from the unlabeled data.

References

[Arjovsky and Bottou, 2017] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *CoRR*, abs/1701.04862, 2017.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[Bahdanau *et al.*, 2016] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*, 2016.

[Bowman *et al.*, 2016] Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. *CoRR*, abs/1603.06021, 2016.

[Chambers and Jurafsky, 2008] Nathanael Chambers and Daniel Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, 2008.

[Chen *et al.*, 2016a] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. In *ACL*, 2016.

[Chen *et al.*, 2016b] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *NIPS*, 2016.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv:1409.1259*, 2014.

[Denton *et al.*, 2015] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, pages 1486–1494, 2015.

[Donahue *et al.*, 2016] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

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

[Gregor *et al.*, 2015] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[Hermann *et al.*, 2015] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NIPS*, pages 1684–1692, 2015.

[Hill *et al.*, 2015] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *ICLR*, 2015.

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

[Huang *et al.*, 2013] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd CIKM*, pages 2333–2338. ACM, 2013.

[Jang *et al.*, 2016] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv*, 2016.

[Jozefowicz *et al.*, 2015] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *Proceedings of ICML-15*, pages 2342–2350, 2015.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

[Kiros *et al.*, 2015] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *NIPS*, 2015.

[Kusner and Hernández-Lobato, 2016] Matt J. Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. *CoRR*, abs/1611.04051, 2016.

[Lee *et al.*, 2015] Moontae Lee, Xiaodong He, Wen-tau Yih, Jianfeng Gao, Li Deng, and Paul Smolensky. Reasoning in vector space: An exploratory study of question answering. *arXiv preprint arXiv:1511.06426*, 2015.

[Lin and Och, 2004] Chin Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. *ACL*, 2004.

[Luong *et al.*, 2015] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *2015 EMNLP*, 2015.

[Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David Mcclosky. The stanford corenlp natural language processing toolkit. In *ACL: System Demos*, 2014.

[Mostafazadeh *et al.*, 2016] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. *Proceedings of NAACL HLT*, 2016.

[Rajpurkar *et al.*, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv:1606.05250*, 2016.

[Richardson *et al.*, 2013] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, 2013.

[Santos *et al.*, 2016] Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *arXiv*, 2016.

[Seo *et al.*, 2016] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.

[Trischler *et al.*, 2016] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *arXiv:1611.09830*, 2016.

[Wang *et al.*, 2016a] Bingning Wang, Shangmin Guo, Kang Liu, Shizhu He, and Jun Zhao. Employing external rich knowledge for machine comprehension. In *IJCAI*, pages 2929–2935, 2016.

[Wang *et al.*, 2016b] Bingning Wang, Shangmin Guo, Kang Liu, Shizhu He, and Jun Zhao. Employing external rich knowledge for machine comprehension. In *IJCAI*, 2016.

[Wang *et al.*, 2016c] Bingning Wang, Kang Liu, and Jun Zhao. Inner attention based recurrent neural networks for answer selection. In *ACL*, 2016.

[Weston *et al.*,] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv*.

[Weston *et al.*, 2015] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: a set of prerequisite toy tasks. *arXiv*, 2015.

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

[Yu *et al.*, 2017] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.