

An Auxiliary Classifier Generative Adversarial Framework for Relation Extraction

Yun Zhao

Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106 USA
yunzhao@cs.ucsb.edu

Abstract

Relation extraction models suffer from limited qualified training data. Using human annotators to label sentences is too expensive and does not scale well especially when dealing with large datasets. In this paper, we use Auxiliary Classifier Generative Adversarial Networks (AC-GANs) to generate high-quality relational sentences and to improve the performance of relation classifier in end-to-end models. In AC-GAN, the discriminator gives not only a probability distribution over the real source, but also a probability distribution over the relation labels. This helps to generate meaningful relational sentences. Experimental results show that our proposed data augmentation method significantly improves the performance of relation extraction compared to state-of-the-art methods.

Introduction

Relation extraction aims to predict attributes and relations for entities in a sentence, which plays an essential role in information extraction (Allahyari et al. 2017). Relation extraction models like Automatic Content Extraction (ACE) (Dodington et al. 2004) use supervised learning methods, which suffer from limited high-quality training data. Traditional supervised approaches utilize human-labeled data, the quantity of which is far from being enough. Labeling sentences using human label is too expensive and not scalable especially when confronted with large datasets like NYT-Freebase (Riedel, Yao, and McCallum 2010). Alternative paradigms are weakly-supervised learning methods such as distant supervision (Mintz et al. 2009). However, distant supervision is noisy, which could result in incorrect labeling. There are also some recent studies on Convolutional Neural Network (CNN) models for relation extraction (Zeng et al. 2014; Zeng et al. 2015), among which Piecewise Convolutional Neural Network (PCNN) based methods (Zeng et al. 2015) represent the state of the art.

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) and their variants such as CGAN (Mirza and Osindero 2014), InfoGAN (Chen et al. 2016) and AC-GAN (Odena, Olah, and Shlens 2017) are appearing as very promising techniques for data generation, especially in

the Computer Vision domain (Radford, Metz, and Chintala 2015). Text generation has not achieved equal success due to its discrete property in words. However, recent studies have proposed several approaches to deal with this problem. A reinforcement learning (RL) algorithm is used to upgrade the generator with reward signals instead of back-propagation in (Yu et al. 2017; Li et al. 2017). Continuous samples are generated by manipulating the temperature of the softmax function and annealing to discrete values via the training process in (Kusner and Hernandez-Lobato 2016; Jang, Gu, and Poole 2016).

In the AC-GAN architecture, class label information was added to both the generator and the discriminator (Odena, Olah, and Shlens 2017). The objective function also has two parts: the cost of the real source and the cost of the correct class. Using AC-GAN as an unsupervised learning approach could solve the lack of training data problem by accurately generating relational sentences. The key contributions of this paper include:

- 1) We propose an end-to-end AC-GAN based relation extraction framework, providing more high-quality training data for relation classifiers.
- 2) Our generator network outputs the positions of the subject and object entities of the Resource Description Frameworks (RDFs), in addition to the next token every time step.
- 3) Experiment results show that our proposed data augmentation method improves the area under the curve (AUC) of the precision recall (PR) curve by 7.66% compared to a PCNN, on NYT-Freebase.

AC-GAN for Relation Extraction

Our model consists of a generator and a discriminator, shown in Fig. 1. The generator is a Long Short Term Memory (LSTM) network with an embedding layer. The discriminator is a CNN based relation extractor, which outputs both a probability distribution over the real source and a probability distribution over the relation labels. To deal with the discrete value issue for GAN in Natural Language Processing (NLP), our sequence generation process is modeled as a sequential decision making process similar to SeqGAN (Yu et al. 2017). The generator is treated as the agent of RL: The state is the generated tokens so far. The action is the next to-

ken to be generated. The reward is the estimated probability of the generated relational sentence being real and good to guide the generator. Note that the discriminator can only estimate reward of a finished sequence, so Monte Carlo Search is employed to obtain the average reward for current state.

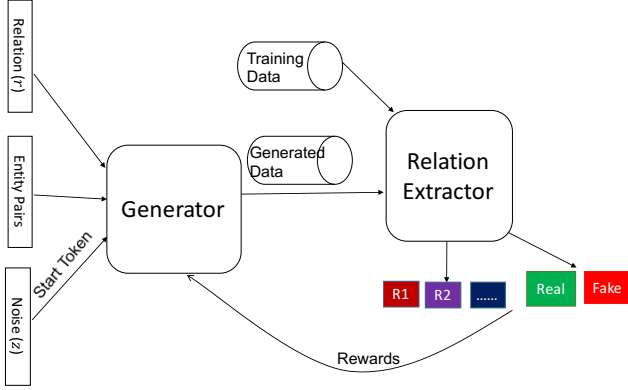


Figure 1: AC-GAN framework for relation extraction.

Generator

The generator is a Recurrent Neural Network with three output layers, as is shown in Fig. 2. In addition to the next token generated by decoders every time step, we introduce two more multilayer perceptron (MLP) layers which regress to the positions of the subject and object entities of the RDFs. For example, the generator generates the sentence of 'Kobe Bryant was a player in Los Angeles Lakers', which combines the information from the three outputs: output tokens, entity1 position (e_{1p}) and entity2 position (e_{2p}). The network structure is inspired by the Faster R-CNN method of object detection in Computer Vision (Ren et al. 2015). The generator exhibits multi-task learning by hard parameter sharing among the three output branches. The losses from the three branches are added together and optimized.

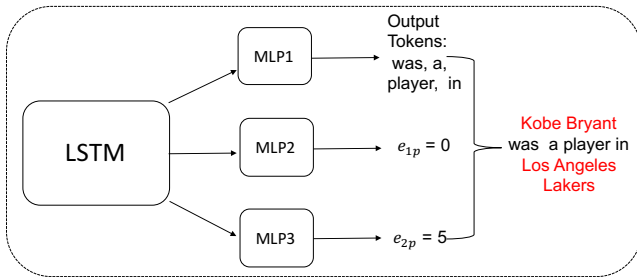


Figure 2: Generator model.

Discriminator

We build the discriminator on existing relation extractor, PCNN (Zeng et al. 2015). Since the generator needs rewards from discriminator to update, the discriminator gives both a probability distribution over the real source and a probability distribution over the relation labels. The loss function L_D

consists of the log-likelihood of the real source, L_S , and the log-likelihood of the correct relation label, L_R .

$$L_D = L_S + L_R. \quad (1)$$

$$L_S = E[\log P(S = \text{real}|X_{\text{real}})] + E[\log P(S = \text{fake}|X_{\text{fake}})]. \quad (2)$$

$$L_R = E[\log P(R = r|X_{\text{real}})] + E[\log P(R = r|X_{\text{fake}})]. \quad (3)$$

Pretraining

We give a warm start to the optimization by pretraining both the generator and the discriminator, since deep RL training is difficult to converge. The generator is pretrained with cross-entropy loss using the RDFs and sentences from the training dataset. The discriminator is pretrained using both real training data and sentences generated by the pretrained generator.

Adversarial Training

In our AC-GAN, every generated sentence has a relation label $r \sim p_r$, in addition to the noise z . GANs deal with a min-max game between generator and discriminator:

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]. \quad (4)$$

The generator tries to fool the discriminator by maximizing the expected reward gotten from the discriminator:

$$J_\theta = E[R_T | s_0, \theta] = \sum_{t=1}^T \sum_{y_t \in y} G_\theta(y_t | s_{t-1}) \cdot Q_{D_\phi}^{G_\theta}(s_{t-1}, y_t). \quad (5)$$

According to REINFORCE algorithm (Sutton and Barto 1998), the gradient of J_θ is approximated using the likelihood ratios:

$$\begin{aligned} \nabla_\theta J_\theta &\approx \sum_{t=1}^T \sum_{y_t \in y} \nabla_\theta G_\theta(y_t | s_{t-1}) \cdot Q_{D_\phi}^{G_\theta}(s_{t-1}, y_t) \\ &= \sum_{t=1}^T \sum_{y_t \in y} G_\theta(y_t | s_{t-1}) \nabla_\theta \log G_\theta(y_t | s_{t-1}) \cdot Q_{D_\phi}^{G_\theta}(s_{t-1}, y_t) \\ &= \sum_{t=1}^T E_{y_t \in G_\theta(s_{t-1}, y_t)} [\nabla_\theta \log G_\theta(y_t | s_{t-1}) \cdot Q_{D_\phi}^{G_\theta}(s_{t-1}, y_t)]. \end{aligned} \quad (6)$$

The reward is generated by multiplying the probability of the corresponding relation and being real.

$$Q_{D_\phi}^{G_\theta}(s_{t-1}, y_t) = R_{\text{relation}} \cdot R_{\text{real}}, \quad (7)$$

where R_{relation} denotes the probability for corresponding relation and R_{real} is the probability of the generated sentences being regarded as real by the discriminator.

Table 1: NYT-Freebase dataset statistics

# Relations (including NA)	53
# Training sentences	570088
# Testing sentences	172448

Table 2: Hyperparameter settings

Generator	
Batch size	64
Adam learning rate	10^{-3}
Relation embedding size	50
LSTM hidden dimension	120
Sequence length	100
Scheduled sampling threshold	0.5
Gradient clip threshold	5.0
Discriminator	
Batch size	64
Adam learning rate	10^{-4}
Word embedding size	50
Position embedding size	5
# Filters	128
Filter window size	3
Dropout keep probability	0.5
Rollout	
Rollout number	6

Experiments

Data

We evaluate our framework on a widely used dataset that was developed by (Riedel, Yao, and McCallum 2010). As is shown in Table 1, the dataset consists of 52 unique relations along with an 'NA' relation. The training dataset contains 570088 sentences, while the testing data has 172448 sentences. We filter out the extra long sentences in addition to the 'NA' labeled sentences to train the generator network. When generating samples we limit the generator to sample sentences for 'non-NA' sentences. To show the effectiveness of the generator, we use Semeval-2010 task 8 dataset (Hendrickx et al. 2009), which contains 8000 semantic relation sentences about 9 relations.

Implementation Details

The word embeddings are from (Lin et al. 2016), with embedding size of 50. The hidden dimension of the LSTM generator is 120. Both the discriminator and the generator are trained with Adam optimizer (Kingma and Ba. 2014) but slightly different learning rates ($lr_{dis} = 10^{-4}$ and $lr_{gen} = 10^{-3}$). When generating rewards for the generated tokens, we use Monte Carlo Search and take the average value of the rewards over rolling out for six times. We implement the deep learning models using Tensorflow (Abadi et al. 2016). More hyperparameter settings are described in Table 2.

During adversarial training, we use teacher forcing similar to (Jang, Gu, and Poole 2016). When the generator receives low rewards at the beginning of adversarial training, it is insufficient to update the generator purely based on the

rewards. The generator needs to see more real training data. In teacher forcing, we apply maximum likelihood estimation (MLE) principle to update the generator more smoothly.

The training process consists of the following parts:

- Pretrain the generator using MLE on training dataset.
- Generate sentences using the pretrained generator.
- Pretrain the discriminator using the generated data and the real training data.
- Jointly train the generator and the discriminator.

Held-out Evaluation on NYT-Freebase

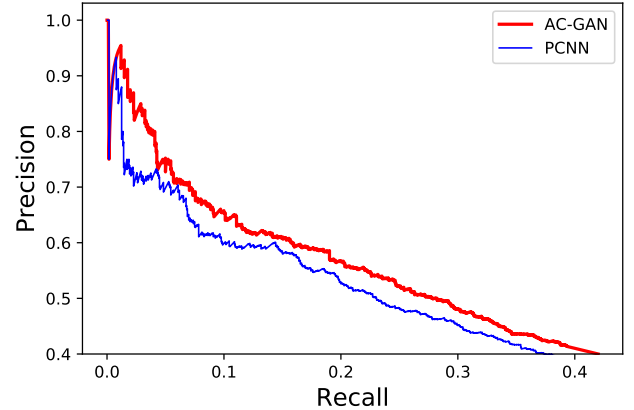


Figure 3: Performance comparison of proposed AC-GAN framework with PCNN.

We compare our proposed AC-GAN framework with PCNN on NYT-Freebase dataset. Similar to previous works (Zeng et al. 2015; Lin et al. 2016), we evaluate the AC-GAN framework using the held-out evaluation. In Fig. 3, experiment results indicate that AC-GAN consistently outperforms PCNN. Our proposed data augmentation method improves the AUC of PR curve by 7.66% compared to PCNN. This is mainly due to the fact that using AC-GAN, the generator can consistently provide high-quality relational sentences.

Generated Samples from AC-GAN trained on Semeval-2010 Task 8 Dataset

Below are some sample relational sentences generated through AC-GAN trained on Semeval-2010 task 8 dataset. Sentences generated with and without discriminator are presented for comparison. We present the Cause-Effect and Product-Producer relation samples in the following.

Cause-Effect Results from AC-GAN:

- The Peru earthquake triggered Avalanche.
- The vascular dilatation was caused by the course.
- Poverty is caused by administration.
- The cysts are caused by every kind of environment.
- The course on a calm day led to more persons with motivation to run gains from long running.

Cause-Effect Results from LSTM only:

The high humidity caused by the bacteria damaged.

The output voltage swing obtainable from the anger is caused by commission from the mug and a huge problem in China.

The decline has always lowing, to take feather samples.

Many more resources are directly the fifth century were almost easy to identify and the late 1990s.

His highly original and topical act disorder encouraged the drama has fever from constipation whereas tricks and sent liable for the Jewish people.

Product-Producer Results from AC-GAN:

This comment refers to this dissertation proposal.

Top problems can center attacks.

Here was the author of the book.

This comment refers to the almost invisible year of Ste90.

A company manufactures books in the story of 85.

The message was generated by the scientist.

Product-Producer Results from LSTM only:

In 1890, the Banbury has constructed a valid plan beans, the human face, from in order being with their responsibilities. The 22-year-old rapper posted a message on his stems that Christmas.

The authors who center the wrong book is 3000 well as a method of two call out of Christmas.

The 22-year-old rapper posted a hill.

Toda has dropped the formal announcement.

The man has dropped the townhouse.

Other relations (like Instrument-Agent etc.) have similar results. It's obvious that sentences generated from AC-GAN are more meaningful compared with sentences generated from LSTM, which could help in relation extraction.

Conclusion

In this paper, we propose an AC-GAN based framework to generate meaningful relational sentences for relation classifiers. Experiment results show that the AC-GAN framework significantly improves relation extraction performance.

References

- [Abadi et al. 2016] Abadi, M.; Barham, P.; Chen, J.; and et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*, 265–283.
- [Allahyari et al. 2017] Allahyari, M.; Pouriyeh, S.; Assefi, M.; Safaei, S.; Trippe, E. D.; Gutierrez, J. B.; and Kochut, K. 2017. A brief survey of text mining: Classification, clustering and extraction techniques. In *arXiv preprint arXiv:1707.02919*.
- [Chen et al. 2016] Chen, Xi, D. Y.; Houthoofd, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, 2172–2180.
- [Doddington et al. 2004] Doddington, G. R.; Mitchell, A.; Przybocki, M. A.; Ramshaw, L. A.; M. Strassel, S.; and Weischede, R. M. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec, vol. 2*, 1.
- [Goodfellow et al. 2014] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.
- [Hendrickx et al. 2009] Hendrickx, I.; Nam Kim, S.; Kozareva, Z.; Nakov, P.; Saghdha, D.; Pad, S.; Pennacchiotti, M.; Romano, L.; and Szpakowicz, S. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, 94–99. Association for Computational Linguistics.
- [Jang, Gu, and Poole 2016] Jang, E.; Gu, S.; and Poole, B. 2016. Adversarial learning for neural dialogue generation. In *arXiv preprint arXiv:1611.01144*.
- [Kingma and Ba. 2014] Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*.
- [Kusner and Hernandez-Lobato 2016] Kusner, M. J., and Hernandez-Lobato, J. M. 2016. Gans for sequences of discrete elements with the gumbel-softmax distribution. In *arXiv preprint arXiv:1611.04051*.
- [Li et al. 2017] Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; and Jurafsky, D. 2017. Adversarial learning for neural dialogue generation. In *arXiv preprint arXiv:1701.06547*.
- [Lin et al. 2016] Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances. In *ACL (1)*.
- [Mintz et al. 2009] Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 1003–1011. Association for Computational Linguistics.
- [Mirza and Osindero 2014] Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. In *arXiv preprint arXiv:1411.1784*.
- [Odena, Olah, and Shlens 2017] Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2642–2651.
- [Radford, Metz, and Chintala 2015] Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. In *arXiv preprint arXiv:1511.06434*.
- [Ren et al. 2015] Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.
- [Riedel, Yao, and McCallum 2010] Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their men-

tions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 148–163.

[Sutton and Barto 1998] Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

[Yu et al. 2017] Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[Zeng et al. 2014] Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network.

[Zeng et al. 2015] Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Emnlp*, 1753–1762.