Using GANs for Predicting Human Choice in Persuasion Games

Gur Keinan and **Idan Pipano**
{gur.keinan, idan.pipano}@campus.technion.ac.il

Abstract

Predicting human behavior in language-based persuasion games has gained substantial attention recently. Existing literature has primarily focused on simulating games to create new data resembling that of real human interactions. We introduce a novel approach using Generative Adversarial Networks (GANs) to directly generate data from the same distribution as real human data, without simulating games. Specifically, we apply a GAN framework to generate interaction data between decision makers and experts in a hotel recommendation game, aiming to predict human choices based on past interactions. Our method leverages GANs to create "realistic" data and assesses its impact on model performance. Despite thorough experimentation, our findings indicate no significant improvement over existing methods, highlighting the complexity and challenges of GAN training. We discuss the implications, limitations, and potential areas for future research, highlighting the necessity for further investigation into the stability and efficacy of GANs in this context.

1 Introduction

Predicting human behaviour in language-based persuasion games has garnered an increasing amount of attention in the last few years (Apel et al., 2022; Shapira et al., 2023). While the recent literature has explored the technique of *simulating* games to obtain new data that aims to represent data of real humans playing the game, we propose an approach which is more direct. We do not simulate games, but rather directly train a Generative Adversarial Network (hereinafter, GAN) to *generate* data from the same distribution of the data of real humans.

The game in which we try to predict human behavior was introduced by (Apel et al., 2022). The game is an interaction between two players: a decision maker (DM) and an expert (travel agent). Each interaction between the DM and the agent consists of multiple rounds, where in each round the agent recommends a hotel to the DM, along with a review of the hotel written by a past costumer, which consists of a score between 0 to 10 and a verbal component. Based on this new information and the previous rounds with the agent, the DM chooses an action - to go to the hotel or not. The goal of the DM is to maximize her utility, which is the number of times she went to a "good" hotel (mean score of 8 or higher) or avoided from going to a "bad" hotel (mean score less than 8). The agent's utility is given by the number of times the DM decided to go to the recommended hotel. Note that this game is a language-based persuasion game which is non-cooperative (the two players do not cooperate) and not zero-sum (the sum of the utilities is not constant).

The rest of the paper is organized as follows: in Section 2 we discuss some further related work, in Section 3, we elaborate on our method of employing GAN to produce persuasion-games data, in Section 4 we detail about the real data we used, and in Section 5 we describe the experiments we conducted and their results.

2 Related Work

Persuasion games Sender-receiver interaction models are central to both economics and Artificial Intelligence, as exemplified by the 2001 Nobel Prize in Economics, which was awarded to Akerlof, Spence, and Stiglitz for pioneering research on this topic (Spence, 1978). A very realistic case within this framework is the case of language-based persuasion games. Language-based games, recently explored in studies such as (Apel et al., 2022; , FAIR; Raifer et al., 2022), are the main focus of this paper. Specifically, we consider a non-cooperative language-based persuasion game, presented in (Apel et al., 2022) and further studied in (Shapira et al., 2023).

GAN's framework The notion of GANs, firstly introduced in (Goodfellow et al., 2014), is a well-known framework for generating data from the same distribution of some given data. In particular, as GANs are known to be prone to unstable training (Goodfellow et al., 2014), a significant amount of work has been conducted to propose various techniques for better training (Salimans et al., 2016; Arjovsky et al., 2017; Gulrajani et al., 2017). There are also works that deal with the difficulties of employing GANs to generate discrete features, as is the case for some of the features we generate, most notably (Jang et al., 2016), who introduced the Gumbel-softmax trick. (Dubiński et al., 2022) proposed a method for selectively choosing the best samples based on certain criteria in order to enhance the training process, which is similar to an approach we take in generating samples, detailed in Paragraph 5.1. Note that while (Dubiński et al., 2022) handpicks generated samples in order to improve the training process, we do it in the aim of enhancing the generating capabilities of an already trained GAN.

3 Model

Following the framework suggested by (Shapira et al., 2023), we try to enrich the human interactions data by training a generative model to imitate the complexities of the data.

We use the GAN framework - training 2 models, a discriminator D and a generator G, in an adversarial manner - the generative model G tries to confuse the discriminator by generating "real - looking" observations while the discriminator aims to distinguish between real and generated data accurately.

Formally, denote the binary cross-entropy loss between the true labels y and predicted labels $\hat{y}$ as $\mathcal{L}(y, \hat{y})$. Then, the loss of the discriminator is

$$\begin{aligned}\mathcal{L}_D = &\mathbb{E}_{x \sim p_{data}}\left[\mathcal{L}\Big(1, D(x)\Big)\right] \\ &+ \mathbb{E}_{z \sim p_z}\left[\mathcal{L}\Big(0, D\big(G(z)\big)\Big)\right]\end{aligned}$$

where p_{data} is the distribution of the real data, and p_z is the distribution of the noise vector used as input to the generator.

The loss of the generator is

$$\mathcal{L}_G = \mathbb{E}_{z \sim p_z}\left[\mathcal{L}\Big(1, D\big(G(z)\big)\Big)\right]$$

In other words, the generator aims to fool the discriminator to classify fake samples as true samples, while the discriminators tries to discriminate correctly between real and fake data.

As mentioned in the introduction section, each interaction between the decision maker (DM) and the agent involves the agent selecting a review (comprising a score and a verbal description) for a predetermined hotel, and the DM deciding whether to visit the hotel or not based on the review. Both parties consider the outcomes of previous interactions in the game.

Instead of directly trying to imitate the full interaction, we use a more subtle approach for speeding up the training and creating more realistic data. First we use the GAN model only for creating the "per-round" data, like the characteristics of the reviews and the choice of the DM. Then, as the data we generated determines all the features, we use the current and past generated data to construct all the features that describe the history of the interaction.

A primary difficulty is that most of the features in a DM-agent interaction are discrete, making it challenging to train a model that samples from their distribution. To overcome this problem we utilize Gumbel noise, a well known method for sampling from discrete distributions in a manner that enables back-propagation. For binary features we use sigmoid activation with Gumbel noise, and for features with multiple values (e.g. ratio of positive to negative review length) we use the Gumbel-Softmax trick.

The architecture we chose for the Generator is an LSTM stacked with a set of fully-connected decision heads. The generation works as follows - in each round the LSTM uses both its output and the constructed features from the last round for creating a new output vector. This vector is then passed through the decision heads to produce the desired features for the current round. This way the generation is past-dependent and hopefully can learn to represent sequential data as we need it to. For the architecture of the Discriminator we chose a simple one-layer fully-connected network, as we observed that an LSTM Discriminator was too strong for the training of the GAN to be effective.

4 Data

In this section we detail about the data used for the GAN training, and then about the data used for training the main model, the model that aims to predict human behaviour in the persuasion game of (Apel et al., 2022).

Data for GAN training We use most of the engineered features (presented in (Apel et al., 2022)) created by GPT 3.5 for embedding the verbal review. We split the interactions to 10-sized blocks and address them as the "real data" for the GAN, the data whose distribution the Generator of our GAN tries to imitate.

Data for main model training We use the human interactions and simulated data as in (Shapira et al., 2023), in proportion of 1:4, the proportion that achieved the best results in (Shapira et al., 2023). Importantly, we also use data generated by our GAN model.

5 Experiments and Results

5.1 Experiment details

First we train our Generator and Discriminator models (to which we refer collectively as the GAN model) on the human training data. We believe it is unnecessary to retrain a GAN model every epoch, so we save the trained models for later use. Then we continue to the main part of the experiment, which is to utilize the new generated data in the existing training loop of the predictive model. We do it by simply adding another phase to every epoch in the training, which uses GAN-generated data. The rest of the training details are exactly as in (Shapira et al., 2023) in order for the results to be properly comparable.

How we use the trained GAN to generate data Traditionally, the way to use a trained GAN to generate new samples is to only use the generator. That is, once the GAN is trained, the discriminator is typically not utilized in the generation phase. Notably, we employ a different approach in the hope of generating samples that better reflect how humans interact in (Apel et al., 2022)'s game. Every time we want to generate a sample from a trained GAN, we use the trained generator to generate **best_of** samples, and then take as our generated sample the sample for which the discriminator gave the highest confidence level of being a real sample.[1] In other words, we take the sample that fooled the discriminator the most.

[1] Recall that the trained discriminator outputs for each sample a confidence level $\in [0, 1]$, with the interpretation that the higher the output, the more the discriminator believes the sample is real and not fake.

%simulations	best_of	mean	epoch
0.5	5	0.835	15
	10	0.836	12
	20	0.837	17
1	5	0.835	21
	10	0.835	22
	20	0.837	19
2	5	0.835	23
	10	0.837	22
	20	0.835	20

Table 1: Results of the main model using GAN-generated data. *mean* - average test accuracy of the model, *epoch* - the number of the epoch which achieved the highest accuracy. The results are based on five seeds.

Hyper-parameters we examine We analyze the impact of the following two hyper-parameters on the training phase: The **best_of** hyper-parameter, which was described above; and the **%simulations** hyper-parameter, the proportion of the generated data compared to the human data. Note that the amount of human data used for training is fixed so **%simulations** effectively determines the amount of generated data.

5.2 Results Analysis and Discussion

Comparision to previous results As depicted in Table 1, our findings do not indicate any notable difference in comparison to the results reported in (Shapira et al., 2023). In Appendix A we present also the confidence intervals we constructed, which show that the difference is not statistically significant. We propose the following explanations:

1. **Training of GANs** - the training of GANs is known to be unstable and difficult, and it is possible that the GAN model we trained did not manage to capture the complexity of the data.

2. **Insufficient data** - during the data preprocessing we needed to remove multiple records that didn't have 10 rounds, and it is possible that the data we used was not sufficient for the GAN to learn the data distribution.

3. **Constructed interaction** - we constructed the interaction in a 10-rounds block. We also experimented with learning the entire interaction in one go (until winning or quitting the game),

but it didn't yield good results. However, it's possible that with more training, better outcomes could have been achieved.

4. **Faulty initial hypothesis** - in the core of our project stands the hypothesis that enriching the data with GAN-generated data will improve the results. It is possible that this hypothesis is simply wrong, and that the combination of the real human-data and (Shapira et al., 2023)'s simulated data is sufficient for predicting human's behaviour in (Apel et al., 2022)'s game. We will, however, not dismiss our hypothesis just yet, as it is well-established that data augmentation techniques—similar to our approach—can enhance outcomes, though it is possible that this specific case may not benefit from it.

Simulations and training effectiveness We point out an interesting relation between the proportion of generated data to human data, **%simulations**, and the epoch in which the best results were achieved. It appears that while comparable optimal results were obtained, the epoch at which these results were achieved is greater as **%simulations** increases. This outcome indirectly indicates that the inclusion of GAN-generated data disrupted the training process, causing the optimizer to exert more effort to locate the optimum. This could be seen as a drawback because the training duration needed to obtain the same performance increased. An alternative view would be that incorporating GAN-generated data imposes regularization effects on the training process.

6 Conclusions

We proposed a novel approach to predicting human behavior in language-based persuasion games using Generative Adversarial Networks (GANs). Our method diverges from the conventional simulation of games by directly training a GAN to generate data from the same distribution as real human interactions. Importantly, we take the approach of using both the trained Discriminiator and the trained Generator to generate data. However, these efforts do not lead to any statistically significant difference in performance of human choice behaviour prediction.

We conclude by stating some key limitations of our work. First, the instability of training a GAN remains a significant challenge, potentially affecting the generated samples and thus the results of our experiments. Second, a comparison which is highly called for but we did not perform due to time constraints is training only with human data and GAN-generated data, without (Shapira et al., 2023)'s simulated data. Moreover, the data we used for training the GAN might have been insufficient. A more comprehensive study is required in order to fully evaluate our proposed method.

References

Reut Apel, Ido Erev, Roi Reichart, and Moshe Tennenholtz. 2022. Predicting decisions in language based persuasion games. *Journal of Artificial Intelligence Research*, 73:1025–1091.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Jan Dubiński, Kamil Deja, Sandro Wenzel, Przemysław Rokita, and Tomasz Trzcinski. 2022. Selectively increasing the diversity of gan-generated samples. In *International Conference on Neural Information Processing*, pages 260–270. Springer.

Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Maya Raifer, Guy Rotman, Reut Apel, Moshe Tennenholtz, and Roi Reichart. 2022. Designing an automatic agent for repeated language–based persuasion games. *Transactions of the Association for Computational Linguistics*, 10:307–324.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Eilam Shapira, Reut Apel, Moshe Tennenholtz, and Roi Reichart. 2023. Human choice prediction in non-cooperative games: Simulation-based off-policy evaluation. *arXiv preprint arXiv:2305.10361*.

Michael Spence. 1978. Job market signaling. In *Uncertainty in economics*, pages 281–306. Elsevier.

A Full Results

In this appendix we present the full results of our main experiment. The results are based on five seeds.

		mean	**epoch**	**CI**
%simulations	**best_of**			
0.5	5	0.835	15	(0.833, 0.837)
	10	0.836	12	(0.834, 0.838)
	20	0.837	17	(0.834, 0.839)
1	5	0.835	21	(0.833, 0.837)
	10	0.835	22	(0.834, 0.836)
	20	0.837	19	(0.834, 0.839)
2	5	0.835	23	(0.833, 0.836)
	10	0.837	22	(0.835, 0.839)
	20	0.835	20	(0.832, 0.837)

Table 2: Results of the main model using GANs data. *mean* - average test accuracy of the model, *epoch* - the number of the epoch which achieved the highest accuracy. The results are based on five seeds.