# DISSECTING AN ADVERSARIAL FRAMEWORK FOR INFORMATION RETRIEVAL

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Recent advances in Generative Adversarial Networks facilitated by improvements to the framework and successful application to various problems has resulted in extensions to multiple domains. IRGAN attempts to leverage the framework for Information-Retrieval (IR), a task that can be described as modeling the correct conditional probability distribution $p(d|q)$ over the documents ($d$), given the query ($q$). The work that proposes IRGAN claims that optimizing their minimax loss function will result in a generator which can learn the distribution, but their setup and baseline term steer the model away from an exact adversarial formulation, and this work attempts to point out certain inaccuracies in their formulation. Analyzing their loss curves gives insight into possible mistakes in the loss functions and better performance can be obtained by using the co-training like setup we propose, where two models are trained in a co-operative rather than an adversarial fashion.

## 1 INTRODUCTION

Information-Retrieval (IR) involves providing a list of ranked documents $\{d_1, d_2, \ldots, d_k\}$ in answer to a query $q$. This general formulation can be extended to various tasks like web-search, where the documents are web pages and information needs are queries, content-recommendation, where the documents are items/content to suggest and queries are users, and Question-Answering, where the documents are answers and queries are questions. The retrieved list can also be viewed as a probability distribution over candidates, one example being $Rank_q(d_i) \equiv p(d_i|q) \propto (\frac{1}{Rank_q(d_i)})^l$, where $l$ is a hyperparameter. Even if the probability distribution is not explicit, it is desirable to retrieve a higher ranked document more often than a lower ranked document.

GANs were proposed as alternatives to generative models and have been shown to be capable of modeling the true data well. High dimensional settings like images and word sequences have seen some success. Given that the generator in GANs tries to model the training data's distribution, adversarial setups seem like a natural fit for IR. The learned distribution can then be used to retrieve relevant documents for incoming queries. IRGAN is a framework proposed by Wang et al. (2017), with the hope of giving Information-Retrieval, access to the large literature of GANs. However, the formulation and implementation of the loss function in the work seems to have a few issues. Specifically, the use of the baseline term recommended in the work results in pitting the loss functions of the discriminator and the generator directly against each other and this leads to issues that are conspicuous in the loss curves.

Given the traction this paper has received since its inception (53 citations as of $27^{th}$ September 2018), it is important to critically analyze the work and attribute the claimed performance improvements correctly. To this end, we propose two models which outperform IRGAN on two of the three tasks and give a comparable performance on the third and serve as an ablation study, by experimentally showing that the generator might not be playing a vital role during train or test time.

The following contributions are made in this work

- We point out inaccuracies in the minimax loss function used in IRGANs
- We substantiate the same by drawing conclusions from the loss curves
- We propose a model motivated by Co-training which outperforms IRGANs

# 2 RELATED WORK

## 2.1 GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks (GANs) (Goodfellow et al. (2014)) were proposed as an alternative to generative models (Salakhutdinov & Larochelle (2010)) which used Markov Chains or other approximations to compute intractable probability distributions. In essence, the *generator* tries to model the real data distribution and the *discriminator* learns to differentiate between real data points and generated data points. GANs are notoriously unstable to train and works like DCGANs (Radford et al. (2015)) and Wasserstein GAN (Arjovsky et al. (2017)) have successfully attempted to alleviate a few issues. Nonetheless, GANs have been widely applied to various problems like image generation, text generation, cross-modal retrieval and more niche ones like Interactive Image Generation (Zhu et al. (2016)), Text to Image (Zhang et al. (2017a)), Image to Image style transfer (Isola et al. (2017)) and robotics (Bousmalis et al. (2017)).

## 2.2 APPLICATIONS TO WORD SEQUENCES

One approach for Sequence modeling is to learn generative models. SeqGAN(Yu et al. (2017)), MaskGAN(Fedus et al. (2018)) and TextGAN(Zhang et al. (2016)) are a few attempts in that direction. SeqGANs are directly related to IRGANs because the generator is trained using a policy gradient, defined by Sutton et al. (2000). The policy-gradient based algorithm allows gradients to be passed back to the generator even though the generation of data involves sampling, unlike in Vanilla GANs (Goodfellow et al. (2014)) where the sample variable ($z$) is part of the input. Though documents are not generated and rather sampled in adversarial frameworks for retrieval, promising results in sequence modeling make it an exciting future prospect to explore the same.

Some novel applications in Natural Language Processing throw some light on how the *discriminator* has an intuitive function. A recent work on question-answering (Joty et al. (2017)) achieves transfer to another language by using the *discriminator* to differentiate between different languages, thus forcing language invariant embeddings. Lample et al. (2017), Zhang et al. (2017b) and Chen et al. (2016) use similar formulations. Another work (Conneau et al. (2017)) achieves word translation without parallel data by working with the hypothesis that different languages have embeddings which are actually linear transformations of each other. This linear transformation, like in the works mentioned before, is learned by using the discriminator to distinguish between the two languages.

## 2.3 RETRIEVAL OF IMAGE RESPONSES

Creswell & Bharath (2016) employed Sketch-GANs for the interesting task of retrieving similar merchant seals (images) based on an input image. DCGANs (Radford et al. (2015)) are used to generate an image, and post training, the last layer of the discriminator is popped off and the rest of it is used as an encoder. This model, however, is specifically for retrieving image responses.

## 2.4 CONDITIONAL GANS

While GANs allow generation based on a random variable $z$, Conditional GANs (Mirza & Osindero (2014)) partition the sample into two parts ($z$ and $y$). $y$ is used to denote which part of the probability distribution the generator has to generate from, and $z$ plays the same role played in Vanilla GANs (Goodfellow et al. (2014)). Figure 1 illustrates the same. Conditional GANs dovetail with IR because $y$ can be used to represent the query or its embedding, and in theory, the model should be able to generate the required document.

$$y \sim query \qquad G(z|y) \sim p_\theta(d|z, q)$$

We feel that an eventual adversarial formulation for IR will be similar to this in flavor.

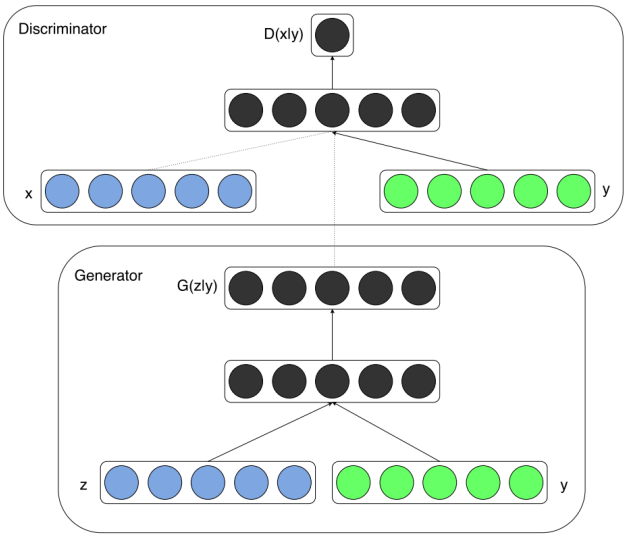



Figure 1: Conditional Generative Nets

## 3 BACKGROUND

In the subsequent sections, $D$ denotes the discriminator, $G$ the generator, $p_{true}$ the real probability distribution over documents, $\phi$ the parameters of the discriminator, $\theta$ the parameters of the generator, $d$ the document, $q$ the query and $r$ the rank of a document with respect to a query.

The equations used to train the *discriminator* and *generator* in Goodfellow et al. (2014) are the following respectively.

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]$$

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$$

The discriminator minimizes the likelihood of a "generated" data point and maximizes it for a "real" data point, while the generator tries to generate data points which the discriminator thinks is "real". The two models are trained alternatively and the procedure culminates in a generator which is able to produce data which looks like the real data.

## 4 IRGAN FORMULATION

This section elucidates the IRGAN formulation (Wang et al. (2017)). Comments by the authors are in italics (in this section alone), while normal typeface is a paraphrased version of IRGAN. IRGAN is motivated by the combination of two schools of thoughts, the generative retrieval model and the discriminative retrieval model.

### 4.1 DISCRIMINATOR AND GENERATOR

The generative retrieval model $p_\theta(d|q, r)$ tries to sample relevant documents from a candidate pool with the aim of cloning the true probability distribution $p_{true}$. The discriminative retrieval model $f_\phi(q, d)$, which a binary classifier, tries to discriminate between real pairs $(q, d)$ and generated ones.

Two different loss functions IRGAN-Pointwise and IRGAN-Pairwise are proposed.

### 4.2 IRGAN-POINTWISE

This is called so because each data point is independently used to train, unlike in IRGAN-Pairwise where *pairs* of *points* are used. The dataset is expected to have some cue with respect to how often a document is correctly retrieved for a query, if at all.

$$J^{G^*,D^*} = \min_{\theta} \max_{\phi} \sum_{n=1}^{N} (E_{d \sim p_{true}(q_n,r)}[(\log D(d|q_n))] + E_{d \sim p_\theta(d|q_n,r)}[(\log 1 - D(d|q_n))])$$

Note that the generator $G$ can alternately be written as $p_{theta}(d|q_n, r)$, which denotes the modeled probability distribution, and $D(d|q) = \sigma(f_\phi(d, q))$ represents the discriminator's score.

### 4.3 IRGAN-PAIRWISE

In some IR problems the training data may not be a set of relevant documents for each query, but rather a set of ordered document pairs $R_n = [< d_i, d_j > | d_i \succ d_j]$, where $d_i \succ d_j$ means that the first document is more relevant for query $q_n$ than the second document. $o$ represents a real pair $< d_u, d_v >$ and $o'$ represents a generated pair $< d'_u, d'_v >$. The discriminator's goal in this setting is to discriminate between $o$ and $o'$, with $D(o|q) = \sigma(f_\phi(d_u, q) - f_\phi(d_v, q))$

$$J^{G^*,D^*} = \min_{\theta} \max_{\phi} \sum_{n=1}^{N} (E_{o \sim p_{true}(o|q_n)}[(\log D(o|q_n))] + E_{o' \sim p_\theta(o'|q_n)}[(\log 1 - D(o'|q_n))])$$

Note the similarity between this and the previous formula. *The problem with this formula is that $D(o|q)$ is actually supposed to denote the probability that the pair o is from the real data distribution and not the probability that the pair is correctly ranked, as mentioned in the paper.*

### 4.4 OPTIMIZING THE GENERATOR

The generator samples documents from the candidate pool based on its belief (relevance score). This sampling has the downside that the gradients cannot be backpropagated, and policy gradients (Sutton et al. (2000)) have to be used. *As an intuition, the documents can be considered as the arms of a contextual multi-arm bandit (Auer et al. (2002), Lu et al. (2010)), and picking an arm can be viewed as analogous to choosing the document as relevant. The policy discovered gives us the relevance of each document and $-\log(1 - D(d|q))$ is the reward for picking that action/document (d).* Let $J^G$ represent the objective function of the generator that it has to maximize. The policy gradient (REINFORCE) can be written as the following.

$$\nabla_\theta J^G(q_n) = \frac{1}{K} \sum_{k=1}^{K} \nabla_\theta \log p_\theta(d_k|q_n, r) \log(1 + exp(f_\phi(d_k, q_n)))$$

### 4.5 BASELINE TERM

To reduce the variance in REINFORCE, a standard trick is to use the advantage function instead of just the reward. This does not change the optimal parameters.

$$\log(1 + exp(f_\phi(d_k, q_n))) - \mathbb{E}_{p_\theta(o'|q_n)}[\log(1 + exp(f_\phi(d_k, q_n)))]$$

Another baseline term that is suggested for each query is $f_\phi(d_+, q)$, where $d_+$ represents the positive document. This is legal because the term does not depend on the document (action). This is motivated by the expectation of a larger generator score if $f_\phi(d_+, q)$ is large and lower if $f_\phi(d_+, q)$ is low. *This baseline term is used in two of their three tasks and causes the violation of adversarial formulation, as we show in the following section.*

# 5 INSIGHTS INTO IRGAN MINIMAX LOSS FUNCTION

Having shown that the generator can be optimized using REINFORCE, we focus on the loss function and show how the baseline term exacerbates training. We consider Stochastic Gradient Descent updates for ease of illustration. Consider a triple $(q, d_r, d_g)$, where $d_r$ denotes the correct document according to the true distribution and $d_g$ denotes the generated document. The discriminator's updates are in the direction of $\nabla J^D$, with the following definition.

$$J^D = \log D(d_r|q) + \log(1 - D(d_g|q))$$

With the baseline term included, the generator's updates are in the direction of $\nabla J^G$, with the following definition.

$$J^G = \log(1 - D(d_r|q)) + \log D(d_g|q)$$

Since maximizing $\log(1 - z)$ with respect to $z$ is the same as maximizing $-\log z$, we can write the following equivalent loss functions

$$J^D = \log D(d_r|q) - \log D(d_g|q) \qquad J^G = -\log D(d_r|q) + \log D(d_g|q)$$

It is apparent that the discriminator and the generator are optimizing directly opposite loss functions and this detrimental to the performance of the models. We provide experimental proof later that the performance improvements shown in IRGAN are mainly because of the discriminator maximizing the likelihood of the real data and not because of the generator.

# 6 EXPERIMENTAL SETUP

This section describes the datasets, models and hyperparameters.

## 6.1 DATASETS

We conduct experiments on three tasks, Web Search, Item Recommendation and Question Answering, using the same datasets mentioned in IRGAN.

Table 1: Datasets

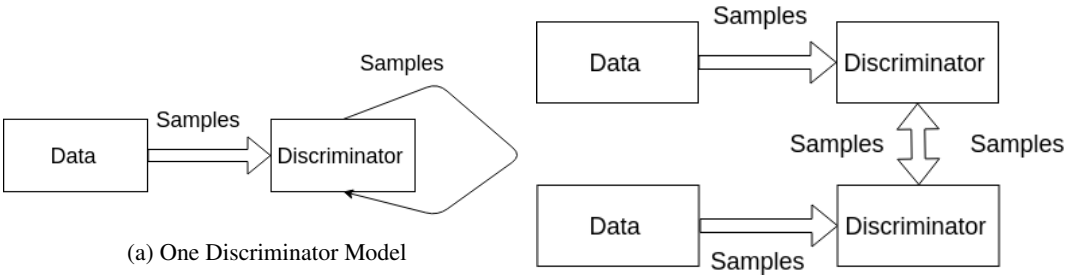
| Task | Dataset |
|---|---|
| Web Search | LETOR by Liu et al. (2007) |
| Item-Recommendation | Movielens |
| Question Answering | InsuranceQA by Feng et al. (2015) |

## 6.2 PROPOSED MODEL

We propose two models to compare and critically analyze performance gains facilitated by IRGAN and illustrate them in Figure 2.

The first model increases the likelihood of the training data and decreases the likelihood of documents which are not relevant to the query but have a high score according to its *own* parameters. It maximizes the following, where the sampling for the second term is from a candidate pool with only negative answers. Not following this will lead to undesirable updates because sampling positive documents for the second term will result in decreasing the likelihood of real data

$$J^{Model1} = \min_{\psi} \max_{\phi} \sum_{n=1}^{N} (E_{d \sim p_{true}(q_n, r)}[(\log D(d|q_n))] - E_{d \sim p_{\psi}(d|q_n, r)}[(\log 1 - D(d|q_n))])$$




(a) One Discriminator Model

(b) Co-training setup

Figure 2: Models

To alleviate the pernicious loss function of IRGAN, we propose a model which uses two discriminators in a co-operative setup influenced by Co-training (Blum & Mitchell (1998)). Instead of using two different views $(x_1, x_2)$ as mentioned in the work, we use the same views for both the discriminators but let them influence each other in a feedback loop. Training is similar to Model 1 with the only difference being that each discriminator decreases the likelihood of documents relevant to the *other* discriminator rather than itself. Co-training is known to help achieve better performance.

### 6.3 Hyperparameters

The hyperparameters for the proposed model are the same except for absence of G_Epochs, for obvious reasons. Information about hyperparameter tuning is mentioned in the Appendix.

Table 2: Hyperparameters for IRGAN

| Hyperparameter | Description |
|---|---|
| Learning Rate | For both generator and discriminator |
| Batch Size | Batch size for training |
| Embed Dim | Embedding dimension of query or document |
| Epochs | Number of epochs of training |
| D_Epochs | Number of epochs the discriminator is trained per epoch |
| G_Epochs | Number of epochs the generator is trained per epoch |
| Temperature | Temperature parameter for softmax sampling of documents |

## 7 Experiments and Discussion

We report only the P@5 and NDCG@5 values because all other metrics follow the same trend.

### 7.1 Web Search

Table 3 reports the performance of various models. As can be seen, both the Single Discriminator and the Co-training models outperform IRGAN models. The fact that each query is associated approximately with $5$ positive documents provides evidence that the proposed models can perform well in sparse reward settings.

### 7.2 Item-Recommendation

This task, in contrast to the other two, has multiple relevant documents that can be retrieved for each query, making it slightly easier. Each user (query) rates a movie (document), and $55\%$ of the entries in the train set and $56\%$ in the test set are relevant pairs. It can be seen in Table 4 that the single discriminator model achieves only a slightly lower score, and given the small size of the dataset ($943$ users), it makes just 7 more mistakes when compared to IRGAN. This is not a statistically significant

Table 3: Results on LETOR dataset used in IRGAN

| Model | P@5 | NDCG@5 |
|---|---|---|
| RankNet (Burges et al. (2005)) | 0.1219 | 0.1709 |
| LambdaRank (Burges et al. (2007)) | 0.1352 | 0.1920 |
| IRGAN-pointwise | 0.1657 | 0.2225 |
| IRGAN-pairwise | 0.1676 | 0.2154 |
| Single Discriminator | 0.1676 | 0.2190 |
| Co-training | **0.1733** | **0.2252** |

number, especially because the IRGAN generator is pre-initialized to a model which scores $0.34$ but our model learns from scratch.

Table 4: Results on Movielens Dataset

| Model | P@5 | NDCG@5 |
|---|---|---|
| BPR (Rendle et al. (2009)) | 0.3044 | 0.3245 |
| LambdaFM (Yuan et al. (2016)) | 0.3474 | 0.3749 |
| IRGAN-pointwise | **0.3750** | **0.4099** |
| Single Discriminator | 0.3675 | 0.3925 |

### 7.3 Question Answering

After close correspondence with the authors of IRGAN, we obtained all the hyperparameters required for the models. Multiple random seeds were used in vain, the results in the paper for Question-Answering tasks could not be replicated. We instead mention the best results out of all random seeds. We believe that if there is some random seed which gives better performance for IRGAN, it should do so for our model as well. The co-training model outperforms IRGAN-Pairwise.

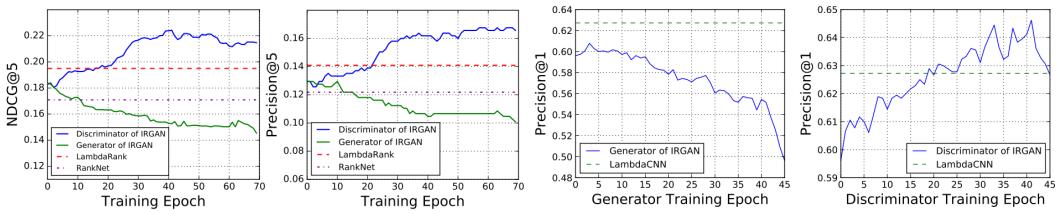
Table 5: P@1 on InsuranceQA

| Model | P@1 |
|---|---|
| IRGAN-Pairwise | 0.616 |
| Single Discriminator | 0.614 |
| Co-training | **0.623** |

### 7.4 Loss curves

The loss curves in Figure 3 picked from IRGAN's work show deteriorating performance of the generator, which is in contrast to what is observed in actual adversarial training. In the minimax setting, since the generator is expected to capture the real data distribution, its performance is supposed to improve and this can indirectly be seen in GANs and DCGANs where the samples generated look more and more like real-world data points. Further, a deteriorating generator implies that the discriminator's improvement in performance is only because of the first term of $J^D$, which hints that our proposed models might be able to do better than IRGAN. The reason offered in the paper is that "A worse generator could be the result of the sparsity of document distribution, i.e., each question usually has only one correct answer". But this reason does not seem plausible, given that DCGANs have been able to model very high dimensional data, where the probability distribution is only a tiny part of the real space.

Further, the increase in performance of the discriminator in all cases is coupled with a deteriorating generator. This substantiates our claim that the discriminator and the generator are optimizing directly opposite loss functions.




(a) Web Search (b) Question Answering

Figure 3: Performance curves

Item-recommendation task is a little different from the other two tasks at hand because of a large number of positive answers. When the loss curves are plotted, though the generator's performance improves, the discriminator's loss remains high and almost constant throughout the procedure, as shown in Figure 4. This is another indication that the performance of IRGAN is not actually because of the adversarial setup, but because of maximization of the likelihood of the real data.

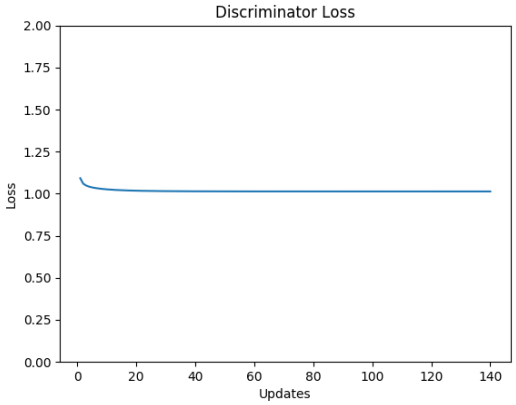



Figure 4: Discriminator Loss for Content-Recommendation

## 8 Conclusion and Future Work

The experiments performed show that IRGAN is by no means state-of-the-art on those datasets. Further, the performance does not justify the large training time of 4 hours per generator epoch and 1 hour of discriminator epoch as opposed to 2 hours per epoch of the co-training model (11 GB GPU and Question Answering task). The shaky mathematical formulation renders the generator useless after training, and any gains in performance can be attributed directly to the first term of $J^D$, where the likelihood of the real data is increased. We showed that the discriminator and generator are optimizing directly opposite loss functions and this is the cause of deleterious training.

The poor performance of IRGAN on Web-Search and Question Answering and only a satisfactory performance on Content-Recommendation (which has dense rewards) lead us to speculate that it does not work well in sparse reward scenarios. This is similar to a well-known problem called the Sparse Reward Reinforcement Learning. We think that a correct formulation along with established techniques from the former, like reward shaping (Ng et al. (1999)) may lead to better performance. Newer methods like Hindsight Experience Replay (Andrychowicz et al. (2017)) which allow models to learn both from mistakes and rewards may further ameliorate learning.

We would also like to explore in the direction of learning correct adversarial frameworks for more complex tasks like Image Retrieval and Question Answering which will involve learning end-to-end trainable models. With advances in modeling sequences, this could also involve generation of documents rather than sampling them.

## REFERENCES


Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pp. 5048–5058, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100. ACM, 1998.

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pp. 7, 2017.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pp. 89–96. ACM, 2005.

Christopher J Burges, Robert Ragno, and Quoc V Le. Learning to rank with nonsmooth cost functions. In *Advances in neural information processing systems*, pp. 193–200, 2007.

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

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

Antonia Creswell and Anil Anthony Bharath. Adversarial training for sketch retrieval. In *European Conference on Computer Vision*, pp. 798–809. Springer, 2016.

William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: Better text generation via filling in the _. *arXiv preprint arXiv:1801.07736*, 2018.

Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. Applying deep learning to answer selection: A study and an open task. *arXiv preprint arXiv:1508.01585*, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

Shafiq Joty, Preslav Nakov, Lluís Màrquez, and Israa Jaradat. Cross-language learning with adversarial neural networks: Application to community question answering. *arXiv preprint arXiv:1706.06749*, 2017.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval*, volume 310. ACM Amsterdam, The Netherlands, 2007.

Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, pp. 485–492, 2010.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pp. 278–287, 1999.

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

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pp. 452–461. AUAI Press, 2009.

Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 693–700, 2010.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 515–524. ACM, 2017.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pp. 2852–2858, 2017.

Fajie Yuan, Guibing Guo, Joemon M Jose, Long Chen, Haitao Yu, and Weinan Zhang. Lambdafm: learning optimal ranking with factorization machines using lambda surrogates. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 227–236. ACM, 2016.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint*, 2017a.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 1959–1970, 2017b.

Yizhe Zhang, Zhe Gan, and Lawrence Carin. Generating text via adversarial training. In *NIPS workshop on Adversarial Training*, volume 21, 2016.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pp. 597–613. Springer, 2016.

# APPENDIX

## HYPERPARAMETERS

The following were the ranges of hyperparameter tuning, along with the best value. Gradient Descent Optimizer was used so that the comparison with IRGAN is fair.

Table 6: Single Discriminator for Web-Search

| Hyperparameter/Seed | Range/List | Best |
|---|---|---|
| Learning Rate | 0.002-0.2 | 0.004 |
| Batch Size | [8,16,32] | 8 |
| Feature Size | [46, 92] | 46 |
| Random Seed | [20,40,60] | 40 |

For the co-training model, for every epoch, we optimize the two discriminators several times. We call these the outer and inner epochs in Table 7.

Table 7: Co-training for Web-Search

| Hyperparameter/Seed | Range/List | Best |
|---|---|---|
| Learning Rate | 0.002-0.2 | 0.006 |
| Outer Epochs | [30,50] | 50 |
| Inner Epochs | [30,50] | 30 |
| Batch Size | [8,16,32] | 8 |
| Feature Size | [46, 92] | 46 |
| Random Seed | [20,40,60] | 40 |

DNS_K in Table 8 represents the number of candidates that are chosen before performing the softmax. This is done to make the procedure computationally tractable. We use the value suggested in IRGAN.

Table 8: Single Discriminator for Content Recommendation

| Hyperparameter/Seed | Range/List | Best |
|---|---|---|
| Learning Rate | 0.01-0.05 | 0.02 |
| Batch Size | 10 | 10 |
| Embedding Dimension | [20, 40, 60] | 20 |
| Random Seed | 70 | 70 |
| DNS_K | 5 | 5 |

Table 9: Single Discriminator for Question Answering

| Hyperparameter | Best |
|---|---|
| Learning Rate | 0.05 |
| Epochs | 20 |
| Batch Size | 100 |
| Embedding Dimension | 100 |

Table 10: Co-training for Question Answering

| Hyperparameter | Best |
|---|---|
| Learning Rate | 0.05 |
| Outer Epochs | 20 |
| Inner Epochs | 1 |
| Batch Size | 100 |
| Embedding Dimension | 100 |