IRGAN: A Minimax Game for Unifying Generative and Discriminative IR Models

SIGIR. 2017

21. 05. 24

Byeong-Hyun Ko

Index

- GAN - Generative Adversarial Nets
- Information Retrieval
- IRGAN
  - Pair-wise extension
  - Training
  - Experiments
    - Web Search
    - Recommend System
    - Question & Answering
- Conclusions

Introduction to GAN (Generative Adversarial Nets)

- Problem Definition in Data Generation
  - Traditional objective: maximum likelihood estimation (MLE)

$$\max_{\theta} \frac{1}{|D|} \sum_{x \in D} [\log q_{\theta}(x)] \simeq \max_{\theta} \mathbb{E}_{x \sim p(x)} [\log q_{\theta}(x)]$$

$D = \{x\}$: Dataset

$q_{\theta}(x)$: Model

$p(x)$: True Distribution (the data we want)

  - Check whether a true data is with a high mass density of the learned model

Introduction to GAN (2)

- Inconsistency of Evaluation and Use

$$\max_\theta \mathbb{E}_{x\sim p(x)}[\log q_\theta(x)]$$

Training/evaluation

- Check whether a true data is with a high mass density of the learned mode

$$\max_\theta \mathbb{E}_{x\sim q_\theta(x)}[\log p(x)]$$

Use

- Check whether a model-generated data is considered as true as possible
- More straightforward but it is hard or impossible to directly calculate p (x)

- **What if we build a discriminator to judge whether a data instance is true or fake (artificially generated)?**

Introduction to GAN (3)

- Generator Network

$$\boldsymbol{x} = G(\boldsymbol{z}; \boldsymbol{\theta})$$

  - Popular implementation: multi-layer perceptron

- Discriminator Network

$$P(\text{true}|\boldsymbol{x}) = D(\boldsymbol{x}; \boldsymbol{\phi})$$

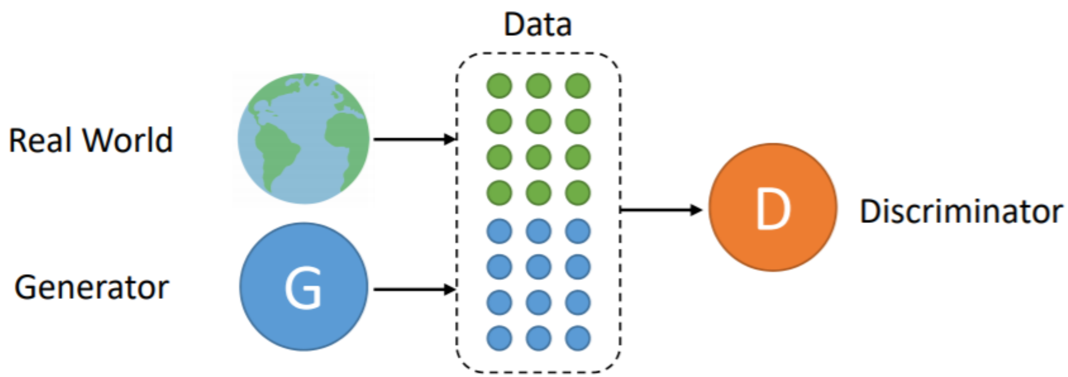
  - Can be implemented by any neural networks with a probabilistic prediction

- GAN Objective Function

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Introduction to GAN (4)

- A Minimax Game

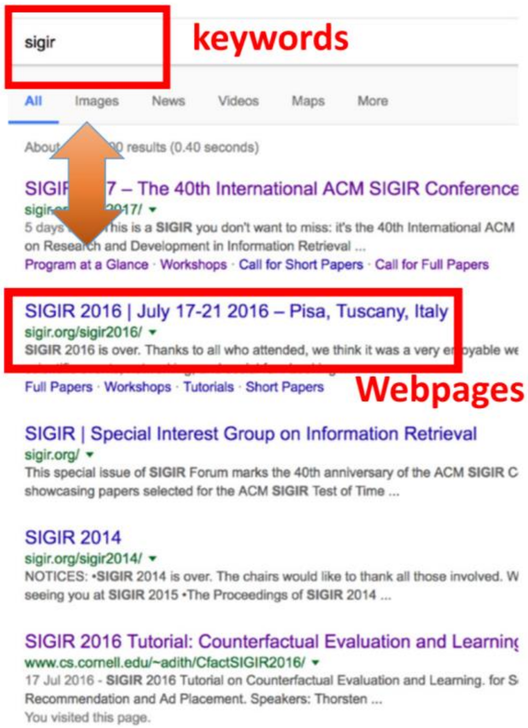


$$\min_G \max_D J(G,D) \qquad \max_D J(G,D)$$

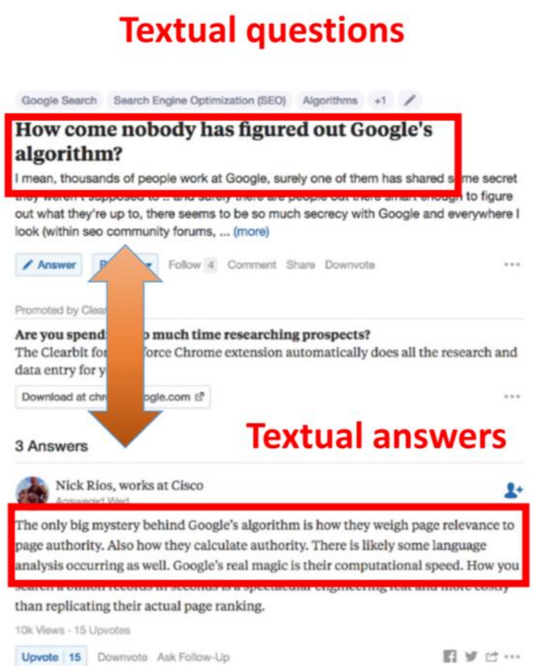
- Discriminator tries to correctly distinguish the true data and the fake model-generated data
- Generator tries to generate high-quality data to fool discriminator
- Untill D cannot distinguish the true and generated data

Information Retrieval (1)

Web Search



Question Answers

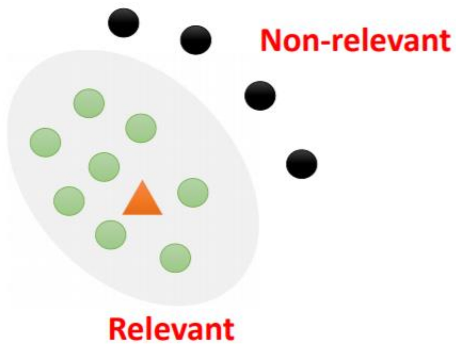


Recommender Systems

- **Relevant** Or not

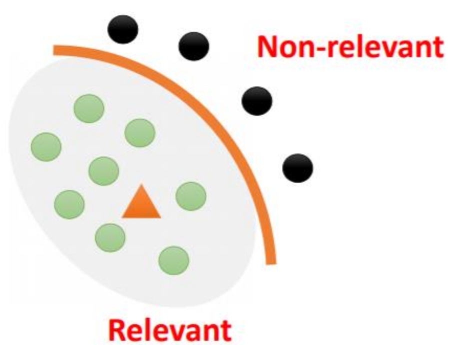
Information Retrieval (2)

- The classic school: Generative Retrieval



  - D -> Q, Q -> D
  - Assume there is an **underlying stochastic generative process** between documents and information needs

- The modern school: Discriminative Model



  - Q + D -> R
  - Discriminative models learned from labeled **relevant judgements or their proxies** such as clicks or ratings

Information Retrieval to IRGAN

Generative models of IR	Discriminative models of IR

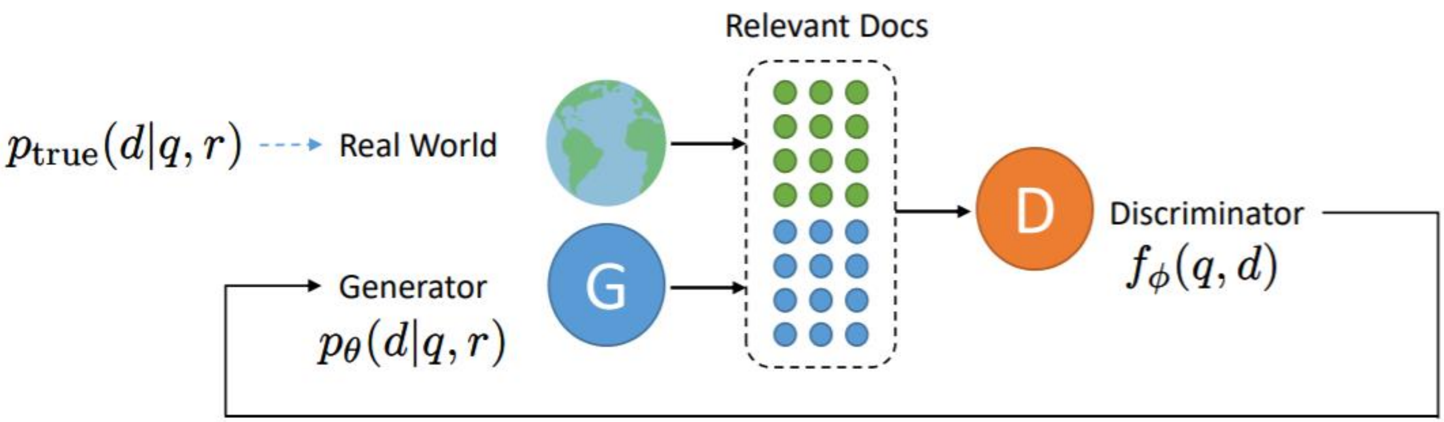
• **Pros:** theoretically sound and very successful in modelling features
• **Cons:**
  • Difficult in leveraging relevancy signals from largely observable data, e.g., links, clicks
  • **Typically not trainable**
	• **Pros:** learn a retrieval ranking function implicitly from labeled data
	• **Cons:** lack a principled way of
	  • Obtaining useful features,
	  • Gathering helpful signals from the massive unlabeled data available, e.g., text statistics, the collection distribution

- If we Mix both advantage
  - Generative models : Learn from discriminative model -> **Trainable !**
  - Discriminative models : Obtain needed training data automatically

IRGAN (1)

- IR + GAN

Generative	Discriminative
Approximate the true relevance distribution (select / generate)	Distinguish between relevant documents or not



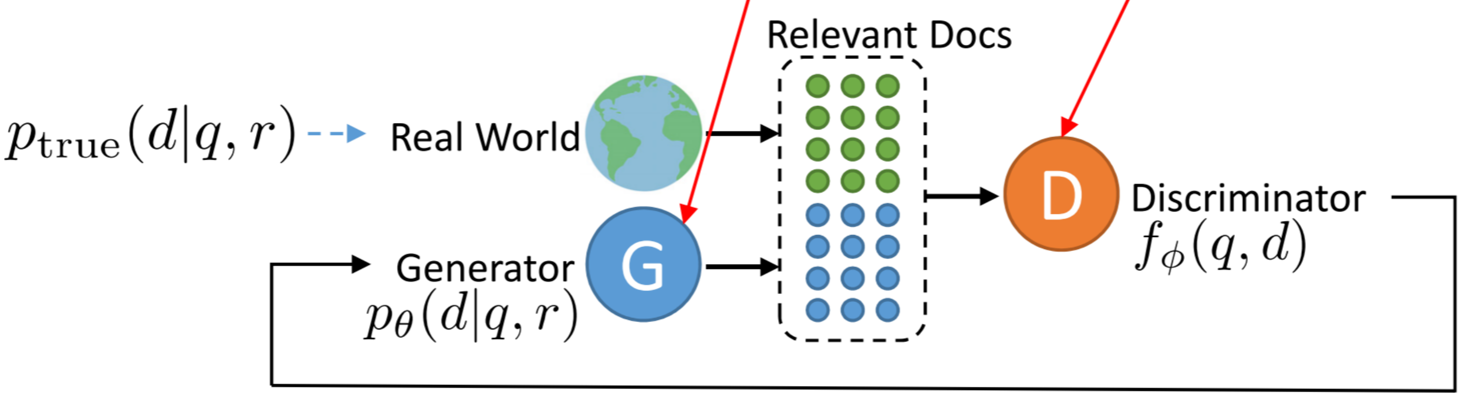
- True relevance distribution : $p_{\text{true}}(d|q,r)$ depicts the user's relevance preference distribution over the candidate documents with respect to query

IRGAN (2)

- IRGAN Objective

$$J^{G^*,D^*} = \min_{\theta} \max_{\phi} \sum_{n=1}^{N} \Big(\mathbb{E}_{d \sim p_{\text{true}}(d|q_n,r)} \left[\log \boxed{D(d|q_n)} \right] +$$

$$\mathbb{E}_{d \sim \boxed{p_\theta(d|q_n,r)}} \left[\log(1 - D(d|q_n)) \right] \Big)$$



* Original GAN : $\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$

IRGAN Extension to Pairwise

- IR problems, it is common that the labelled training data available for learning to rank are not a set of relevant documents
- In the set of ordered document pairs for each query
  - Capture relative preference in pair > Absolute relevance judgements
- Relevance scales > Binary relevance

$$J^{G^*,D^*} = \min_{\theta} \max_{\phi} \sum_{n=1}^{N} \Big(\mathbb{E}_{\mathbf{o} \sim p_{\text{true}}(\mathbf{o}|q_n)} \left[\log D(\mathbf{o}|q_n) \right] + \mathbb{E}_{\mathbf{o}' \sim p_\theta(\mathbf{o}'|q_n)} \left[\log(1 - D(\mathbf{o}'|q_n)) \right] \Big)$$

$$R_n = \{\langle d_i, d_j \rangle | d_i > d_j\}$$
$$\mathbf{o} = \langle d_u, d_v \rangle$$
$$\mathbf{o}' = \langle d'_u, d'_v \rangle$$

IRGAN – Training

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

- Generator Network

$$\phi^* = \arg\max_{\phi} \sum_{n=1}^{N} \Big(\mathbb{E}_{d \sim p_{\text{true}}(d|q_n,r)} \left[\log(\sigma(f_\phi(d,q_n))\right] + \mathbb{E}_{d \sim p_{\theta^*}(d|q_n,r)} \left[\log(1 - \sigma(f_\phi(d,q_n)))\right] \Big),$$

- Discriminator Network

$$\theta^* = \arg\min_{\theta} \sum_{n=1}^{N} \Big(\mathbb{E}_{d \sim p_{\text{true}}(d|q_n,r)} \left[\log \sigma(f_\phi(d,q_n))\right] + \mathbb{E}_{d \sim p_\theta(d|q_n,r)} \left[\log(1 - \sigma(f_\phi(d,q_n)))\right] \Big)$$

$$= \arg\max_{\theta} \sum_{n=1}^{N} \underbrace{\mathbb{E}_{d \sim p_\theta(d|q_n,r)} \left[\log(1 + \exp(f_\phi(d,q_n)))\right]}_{\text{denoted as } J^G(q_n)}$$

IRGAN – Training (2)

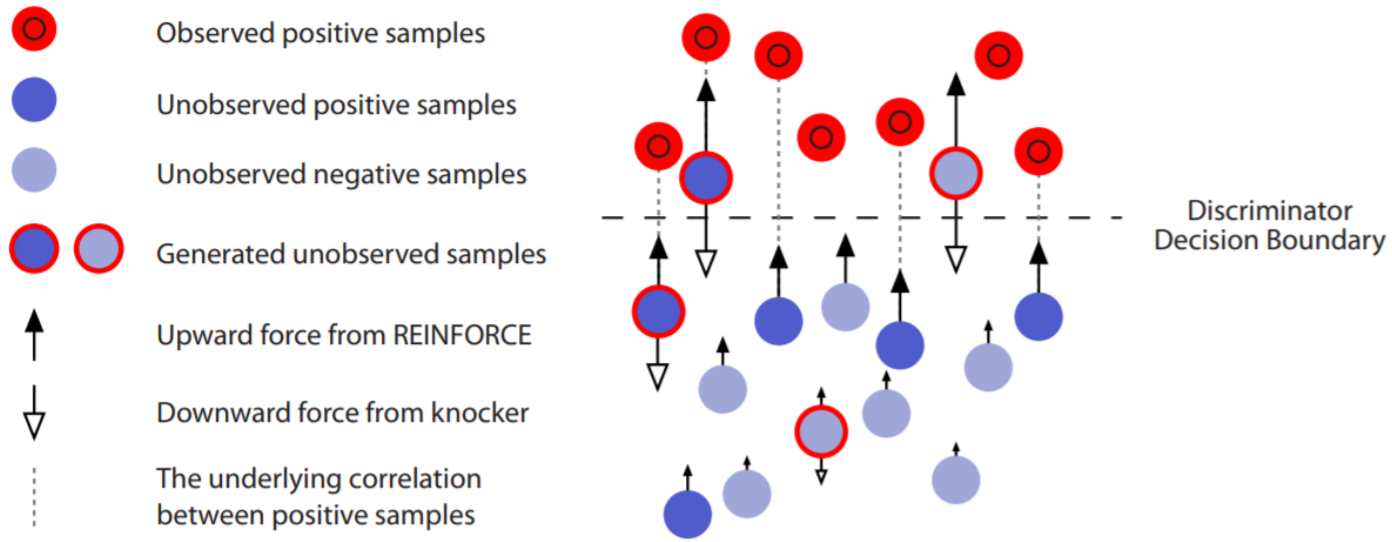


Figure 1: An illustration of IRGAN training.

- Sample -> Soup
- Discriminator Decision Boundary -> Surface of water
- Relevant, Correlation -> floatable soup is fixed in situation (by density)
- Density of Water -> Adjust by Generative & Discriminate term

IRGAN – Experiment (Web-1)

- Scoring Functions : RankNet – 2 layer NN ;

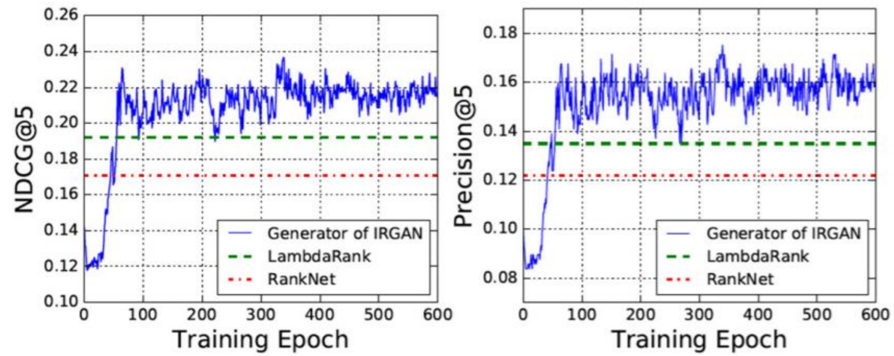
$$s(q,d) = \boldsymbol{w}_2^\top \tanh(\boldsymbol{W}_1 \boldsymbol{x}_{q,d} + \boldsymbol{b}_1) + w_0$$

- Dataset : MQ-2008 (Millionquery Track in LETOR 4.0)
  - Semi-supervised learning: unlabeled query document pairs
- Task : Rank the candidate documents for each query
- Query : 46-dim vec
- Relevance Level : -1, 0, 1, 2

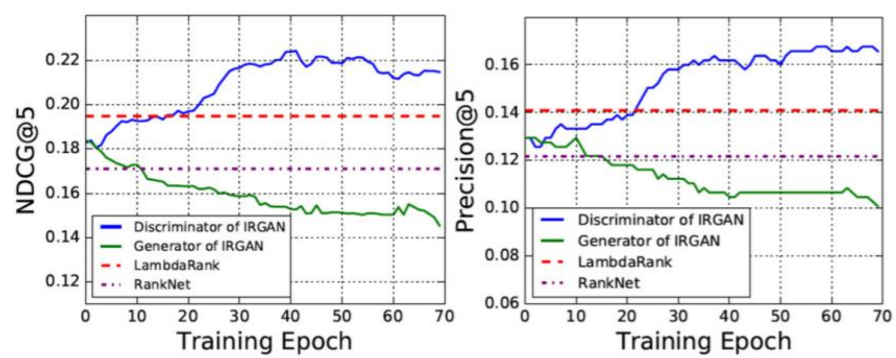
	P@3	P@5	P@10	MAP
MLE	0.1556	0.1295	0.1029	0.1604
RankNet [3]	0.1619	0.1219	0.1010	0.1517
LambdaRank [5]	0.1651	0.1352	0.1076	0.1658
LambdaMART [4]	0.1368	0.1026	0.0846	0.1288
IRGAN-pointwise	0.1714	0.1657	**0.1257**	**0.1915**
IRGAN-pairwise	**0.2000**	**0.1676**	0.1248	0.1816
Impv-pointwise	3.82%	22.56%*	16.82%*	15.50%*
Impv-pairwise	21.14%*	23.96%*	15.98%	9.53%
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.1893	0.1854	0.2054	0.3194
RankNet [3]	0.1801	0.1709	0.1943	0.3062
LambdaRank [5]	0.1926	0.1920	0.2093	0.3242
LambdaMART [4]	0.1573	0.1456	0.1627	0.2696
IRGAN-pointwise	0.2065	**0.2225**	**0.2483**	**0.3508**
IRGAN-pairwise	**0.2148**	0.2154	0.2380	0.3322
Impv-pointwise	7.22%	15.89%	18.63%	8.20%
Impv-pairwise	11.53%	12.19%	13.71%	2.47%

IRGAN – Experiment (Web-2)

- Performance is relied on training epoch
- Typically, when one player (G or D) starts to outperforms the baseline discriminative model, the other player (D or G) would get worse than the baseline

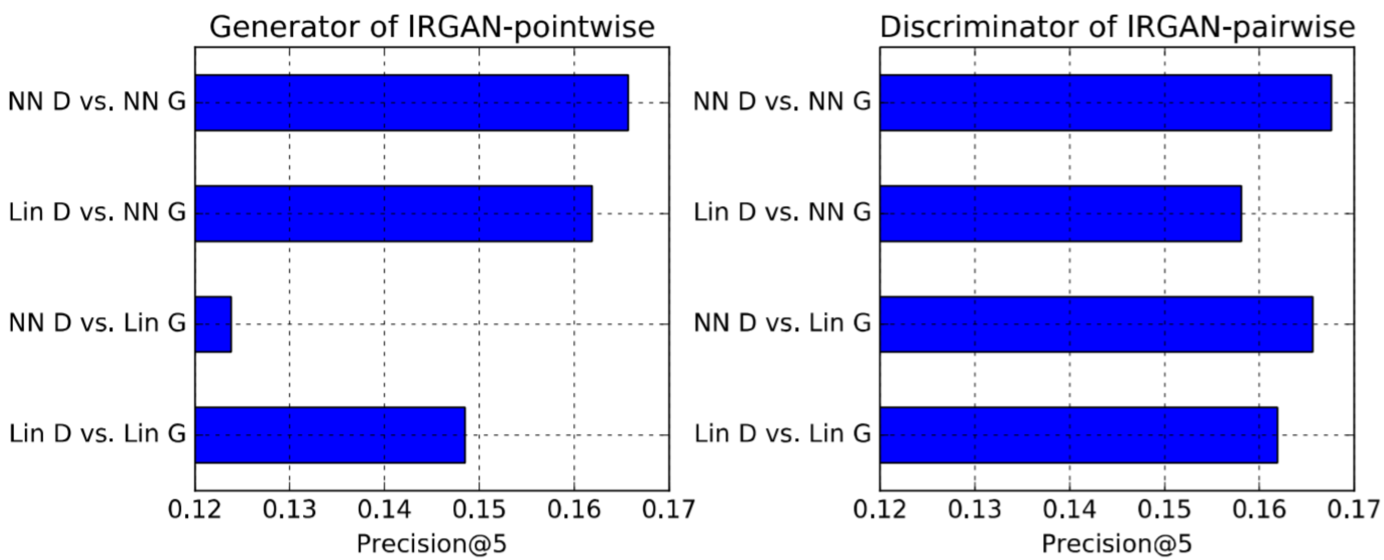


IRGAN-Pointwise Generator Performance



IRGAN-Pairwise Discriminator Performance

IRGAN – Experiment (Web-3)



- IRGAN-pointwise : NN implemented generator works be better than its linear version
  - NN implemented discriminator may not offer a good guidance if the generator has lower model complexity (i.e. linear).
- IRGAN-pairwise : NN implemented discriminator outperforms its linear version
  - Prediction part should be implemented with a capacity at least as high as its opponent.

IRGAN – Experiment (RS)

- Scoring Functions : $s(u,i) = b_i + \boldsymbol{v}_u^\top \boldsymbol{v}_i$
- Dataset :
  - Movielens: 943 users, 1.7k items, 100k
  - Netflix: 480k users, 17k items, 100M
- Task : Top-N item recommendation with implicit feedback data

Table 3: Item recommendation results (Movielens).

	P@3	P@5	P@10	MAP
MLE	0.3369	0.3013	0.2559	0.2005
BPR [34]	0.3289	0.3044	0.2656	0.2009
LambdaFM [46]	0.3845	0.3474	0.2967	0.2222
IRGAN-pointwise	0.4072	0.3750	0.3140	0.2418
Impv-pointwise	5.90%*	7.94%*	5.83%*	8.82%*
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.3461	0.3236	0.3017	0.5264
BPR [34]	0.3410	0.3245	0.3076	0.5290
LambdaFM [46]	0.3986	0.3749	0.3518	0.5797
IRGAN-pointwise	0.4222	0.4009	0.3723	0.6082
Impv-pointwise	5.92%*	6.94%*	5.83%*	4.92%*

Table 4: Item recommendation results (Netflix).

	P@3	P@5	P@10	MAP
MLE	0.2941	0.2945	0.2777	0.0957
BPR [34]	0.3040	0.2933	0.2774	0.0935
LambdaFM [46]	0.3901	0.3790	0.3489	0.1672
IRGAN-pointwise	0.4456	0.4335	0.3923	0.1720
Impv-pointwise	14.23%*	14.38%*	12.44%*	2.87%*
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.3032	0.3011	0.2878	0.5085
BPR [34]	0.3077	0.2993	0.2866	0.5040
LambdaFM [46]	0.3942	0.3854	0.3624	0.5857
IRGAN-pointwise	0.4498	0.4404	0.4097	0.6371
Impv-pointwise	14.10%*	14.27%*	13.05%*	8.78%*

IRGAN – Experiment (QnA)

- Scoring Functions : $s(q,a) = \cos(\boldsymbol{v}_q, \boldsymbol{v}_a) = \frac{\boldsymbol{v}_q^\top \boldsymbol{v}_a}{|\boldsymbol{v}_q| \cdot |\boldsymbol{v}_a|}$
  - Use convolutional layer on embedding matrix of a question sentence or an answer sentence (with MaxPooling)
- Dataset : InsuranceQA Dataset
  - 12k question answer pairs
  - Two test sets with 1.8k pairs
  - rank top-1 answer for each question
- Lower precision in Generator
  - Sparsity : Each Question usually has only one correct answer and many weaker negative answers

Table 5: The Precision@1 of InsuranceQA.

	test-1	test-2
QA-CNN [9]	0.6133	0.5689
LambdaCNN [9, 51]	0.6294	0.6006
IRGAN-pairwise	0.6444	0.6111
Impv-pairwise	2.38%*	1.75%

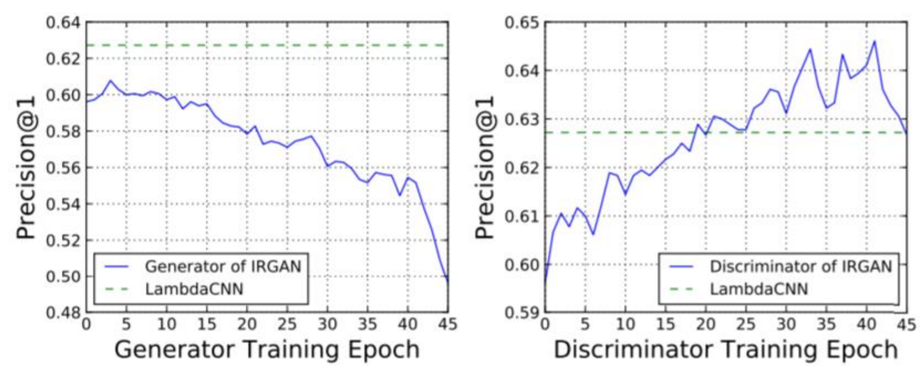


Figure 8: The experimental results in QA task.

Conclusion

- Advantage of Adversarial Model
  - Generator is guided by the signals obtained from the discriminative retrieval mode
  - Discriminator could be enhanced to rank top documents better via strategic negative sampling from the generator
  - IRGAN-> flexible and principled training environment

- Future Work
  - Generalized word token -> applying language model