SIGIR ’17, August 07-11, 2017, Shinjuku, Tokyo, Japan

IRGAN: A MINIMAX GAME FOR UNIFYING GENERATIVE AND DISCRIMINATIVE INFORMATION RETRIEVAL MODELS

Jun Wang
University College London
j.wang@cs.ucl.ac.uk

Lantao Yu, Weinan Zhang*
Shanghai Jiao Tong University
wnzhang@sjtu.edu.cn

Yu Gong,
Yinghui Xu Alibaba Group
renji.xyh@taobao.com

Benyou Wang, Peng Zhang
Tianjin University
pzhang@tju.edu.cn

Dell Zhang Birkbeck,
University of London
dell.z@ieee.org

INTRODUCTION

FORMULATION

- A minimax Retrieval Framework
- Extension to Pairwise Case
- Discussion
- Links to Existing work

APPLICATION

- Web Search
- Item Recommendation
- Question Answering

EXPERIMENTS

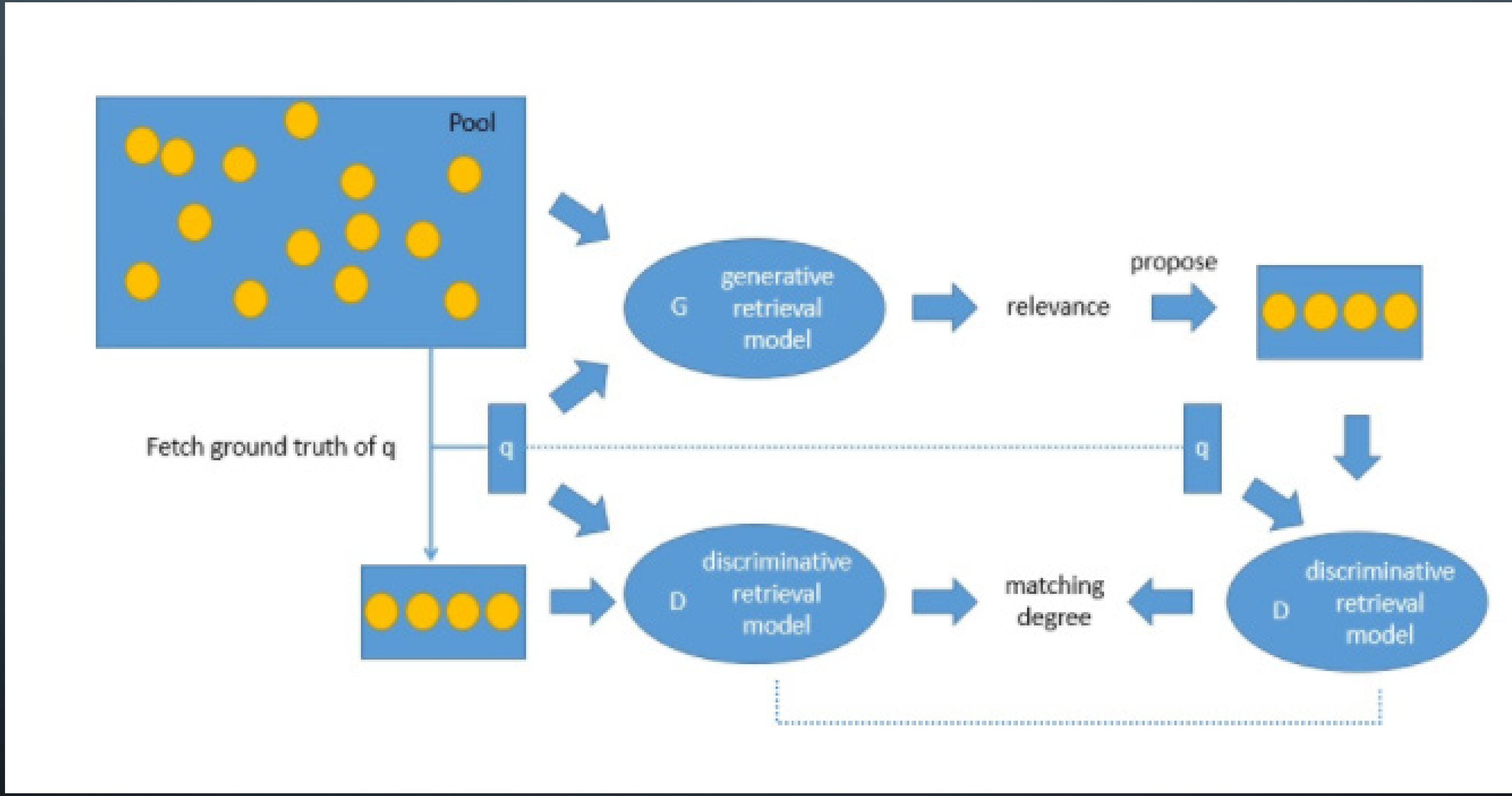
- Web Search
- Item Recommendation
- Question Answering

INTRODUCTION

- Generator solution (relevant) document is generated from a given information need: $q \rightarrow d$ ➡ $P(d \mid q)$
- Statistical language models of text retrieval consider a reverse generative process from a document to a query: $d \rightarrow q$ ➡ $P(q \mid d)$
- Discriminative (classification) solution learned from labelled relevant judgements or their proxies such as clicks or ratings :$q + d \rightarrow r$ ➡ $P(d , q)$

- Discriminative model $p_\phi(r|q,d)$ aims to maximise the objective function by learning from labelled data
- Generative retrieval model $p_\vartheta(d|q,r)$ acts as a challenger who constantly pushes the discriminator to its limit
- Existing approaches generally try to model the interaction between user and system, whereas our approach aims to unify generative and discriminative IR models.

FRAMEWORK



Pool
generative
G
retrieval
model
relevance
propose
Fetch ground truth of q
q
q
discriminative
D
retrieval
model
matching
degree
discriminative
D
retrieval
model

FRAMEWORK

- For a given query q_n ,a set of relevant documents labelled, the size of which is much smaller than the total number of documents M.
- true relevance distribution as conditional probability $p_{true}(d \mid q, r)$, which depicts the (user's) relevance preference distribution over the candidate documents with respect to her submitted query.
- Given a set of samples from $p_{true}(d \mid q, r)$ observed as the training data

Minimax Retrieval Framework

- **Generative retrieval model** $p_\theta(d|q, r)$ which tries to select relevant documents, from the candidate pool for the given query q
- Its goal is to approximate the true relevance distribution $p_{\text{true}}(d|q, r)$ as much as possible.
- **Discriminative retrieval model** $f_\phi(q, d)$, discriminate well-matched query-document tuples (q , d) from ill-matched ones, where the goodness of matching given by $f_\phi(q, d)$, depends on the relevance of d to q
- Its goal is to distinguish between relevant documents and nonrelevant documents for the query q

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

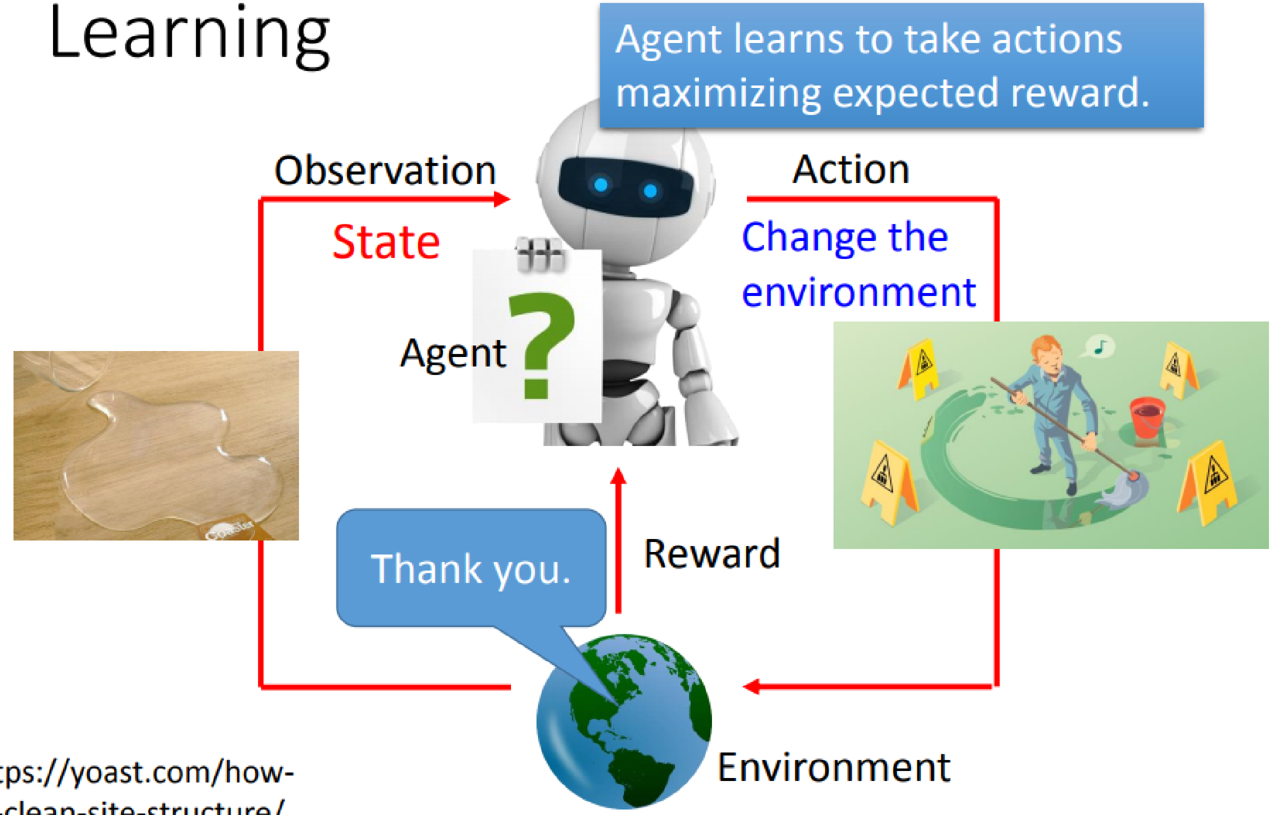
Optimising Discriminative Retrieval

- *maximise* the log-likelihood of correctly distinguishing the true and generated relevant documents.
- With the observed relevant documents, and the ones sampled from the current optimal generative model $p_{\theta^*}(d\,|q, r\ \ )$
- $\phi^* = arg \max_{\phi} \sum_{n=1}^{N} \left(\mathbb{E}_{d \sim p_{true}(d|q_n, r)} \left[\log \sigma \left(f_\phi(d,q) \right) \right] + \mathbb{E}_{d \sim p_\theta(d|q_n, r)} \left[log \left(1 - \sigma \left(f_\phi(d,q) \right) \right) \right] \right)$
- where if the function f_ϕ is differentiable with respect to ϕ, the above is solved typically by stochastic gradient descent

Optimising Generative Retrieval.

- *minimise* the objective;it fits the underlying relevance distribution over documents $p_{\text{true}}(d \mid q, r)$, randomly samples documents from the whole document set in order to *fool* the discriminative retrieval model.

Scenario of Reinforcement Learning



https://yoast.com/how-to-clean-site-structure/

Optimising Generative Retrieval.

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

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

- As the sampling of d is discrete, it cannot be directly optimised by gradient descent as in the original GAN formulation.
- A common approach is to use policy gradient based reinforcement learning
- $\mathbb{E}_{d \sim p_\theta(d|q_n, r)} \left[log \left(1 + exp \left(f_\phi(d, q) \right) \right) \right] = J^G(q_n)$

Optimising Generative Retrieval. policy gradient based reinforcement learning

- $\nabla_\theta J^G(q_n)$

$$= \nabla_\theta \mathbb{E}_{d \sim p_\theta(d|q_n, r)} \left[log \left(1 + exp \left(f_\phi(d, q_n) \right) \right) \right]$$

$$= \sum_{i=1}^{M} \nabla_\theta \, p_\theta \,(d_i | q_n, r) \; log \left(1 + exp \left(f_\phi(d_i, q_n) \right) \right)$$

$$= \sum_{i=1}^{M} p_\theta \,(d_i | q_n, r) \; \nabla_\theta log \, p_\theta \,(d_i | q_n, r) log \left(1 + exp \left(f_\phi(d_i, q_n) \right) \right)$$

$$= \mathbb{E}_{d \sim p_\theta(d|q_n, r)} \left[\nabla_\theta log \, p_\theta \,(d | q_n, r) log \left(1 + exp \left(f_\phi(d, q_n) \right) \right) \right]$$

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

Algorithm 1 Minimax Game for IR (a.k.a IRGAN)

Input: generator $p_\theta(d|q,r)$; discriminator $f_\phi(\mathbf{x}_i^q)$;
training dataset $\mathcal{S} = \{\mathbf{x}\}$

Initialise $p_\theta(d|q,r)$, $f_\phi(q,d)$ with random weights θ, ϕ.
Pre-train $p_\theta(d|q,r)$, $f_\phi(q,d)$ using $\mathcal{S}$
repeat
  for g-steps **do**
    $p_\theta(d|q,r)$ generates K documents for each query q
    Update generator parameters via policy gradient Eq. (5)
  end for
  for d-steps **do**
    Use current $p_\theta(d|q,r)$ to generate negative examples and combine with given positive examples $\mathcal{S}$
    Train discriminator $f_\phi(q,d)$ by Eq. (3)
  end for
until IRGAN converges

- the generator and discriminator can be initialised by their conventional models.
- Eq(5) : $\frac{1}{K}\sum_{k=1}^{K} \nabla_\theta log\ p_\theta\,(d_k|q_n, r)log\left(1 + exp\left(f_\phi(d_k, q_n)\right)\right)$
- Eq(3) : $arg\max_\phi \sum_{n=1}^{N}\left(\mathbb{E}_{d\sim p_{true}(d|q_n,r)}\left[\log\sigma\left(f_\phi(d,q)\right)\right] + \mathbb{E}_{d\sim p_\theta(d|q_n,r)}\left[log\left(1-\sigma\left(f_\phi(d,q)\right)\right)\right]\right)$

Links to Existing Work

- G provides different negative samples to the D training(like negative sampling)
- D's reward signal provides strategic guidance for G training
- G pick documents are regards as negative sample , but pseudo relevance feedback are regards as positive samples
- G will have many iterations , but pseudo relevance feedback usually no further iterations

Web Search

- $x_{q,d} \in R^k$, each dimension represented some value(BM25、TFIDF、PageRank)
- $s(q,d) = w_2^T \tanh\left(W_1 x_{q,d} + b_1\right) + w_0$

Item recommendation

- Matrix factorisation
- s$(u,i) = b_i + v_u^T v_i$
- $b_i =$ basic item , v_u 、v_i =latent vector of user and item

Question Answering

- Question and Answer is represented as a sequence of words
- Using CNN or LSTM to learn sequence of words(l-words), each word is embedded as a vector(k-dim), now we have matrix in $R^{l \times k}$
- After CNN get v_q and $v_a \in$ in R^{Z} , z = number of convolutional kernels.
- $s(q,a) = \cos(v_q, v_a)$

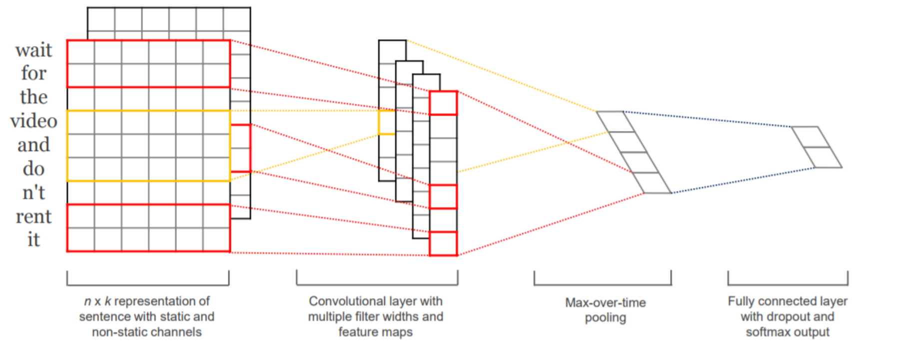


Figure 1: Model architecture with two channels for an example sentence.