# Policy-Gradient Training of Language Models for Ranking

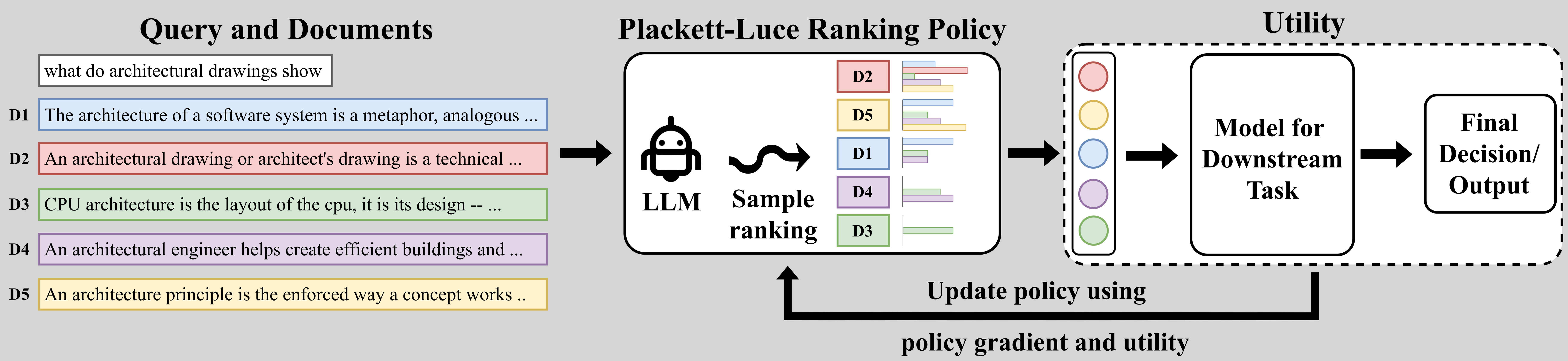
Ge Gao, Jonathan D. Chang, Claire Cardie, Kianté Brantley, Thorsten Joachims

## Overview

How to train LLM-based retrieval models that directly optimize downstream decision-making quality?

We introduce Neural PG-RANK:

- Learns to rank by instantiating a LLM as a Plackett-Luce ranking policy
- End-to-end training of retrieval models as part of larger pipelines via policy gradient
- Can optimize the ranker for any cardinal loss function evaluating the downstream decisions




## Method

Optimize a Plackett-Luce policy $\pi_\theta(r|q) = \prod_{i=1}^{n} \frac{\exp s_\theta(q, d_{r(i)})}{\sum_{j \in \{r(i),\ldots,r(n)\}} \exp s_\theta(q, d_j)}$

+ REINFORCE update
+ Monte Carlo sampling with N samples
+ Variance reduction with leave-one-out baseline
+ nDCG@10 as utility function

$$\widehat{\nabla}_\theta U(\pi_\theta|q) = \frac{1}{N}\sum_i \left[\nabla_\theta \log \pi_\theta(r_i|q)\left(\Delta(r_i|q) - \frac{1}{N-1}\sum_{j\neq i}\Delta(r_j|q)\right)\right]$$

$$= \frac{1}{N}\sum_i \left[\sum_k \nabla_\theta \log \pi_\theta(r_{i,k}|q, r_{i,1:k-1})\right.$$

$$\left.\left(\mathrm{nDCG}(r_{i,k:}|q, r_{i,1:k-1}) - \frac{1}{N-1}\sum_{j\neq i}\mathrm{nDCG}(r_{j,k:}|q, r_{i,1:k-1})\right)\right]$$

## Experimental Setup

Data: MS MARCO for training; BEIR for evaluation

Evaluation metric: nDCG@10

Our ranking policy: either SBERT or TAS-B as warmstart, with Neural PG-RANK method as fine-tuning

Comparison systems: supervised learning SOTA bi-encoder models with distilbert-base-uncased

| Method | Source of Negative Docs: In-Batch | BM25 | Dense Model | Additional Supervision | Loss |
|---|---|---|---|---|---|
| SBERT (Reimers & Gurevych, 2019) | | ✓ | ✓✓✓ | ✓ | MarginMSE + NLL |
| TAS-B (Hofstätter et al., 2021) | ✓ | ✓ | | ✓✓ | MarginMSE + Distillation |
| SPLADEv2 (Formal et al., 2021) | | ✓ | ✓✓ | ✓ | MarginMSE + Sparsity |
| Neural PG-RANK (Ours) | | | ✓ | | Utility Maximization |

## Second-Stage Reranking

Setup: search over a candidate set of 1k documents per query

In-domain Results:

- Performance gains with both warmstart models

Out-of-domain Results:

- Comparable generalization
- Notable improvements on widely-studied QA datasets
- Weaker in the domain of science and finance

| Dataset | Domain | Comparison Systems: SBERT* | TAS-B* | SPLADEv2* | Ours: Neural PG-RANK with SBERT | with TAS-B |
|---|---|---|---|---|---|---|
| MS MARCO dev | misc. | 0.892 | 0.893 | 0.900 | **0.987** | 0.982 |
| TREC-DL 2019 | misc. | 0.743 | **0.749** | **0.749** | 0.742 | 0.741 |
| TREC-COVID | bio-medical | **0.764** | 0.711 | 0.731 | 0.690 | 0.630 |
| NFCorpus | bio-medical | 0.308 | 0.320 | **0.341** | 0.249 | 0.303 |
| NQ | Wikipedia | 0.836 | 0.836 | 0.854 | 0.869 | **0.878** |
| HotpotQA | Wikipedia | 0.747 | 0.785 | 0.834 | **0.902** | 0.900 |
| FiQA-2018 | finance | 0.291 | 0.279 | **0.342** | 0.131 | 0.139 |
| ArguAna | misc. | 0.351 | 0.479 | **0.480** | 0.354 | 0.443 |
| Touché-2020 | misc. | **0.480** | 0.423 | 0.460 | 0.363 | 0.361 |
| Quora | Quora | 0.962 | **0.982** | 0.967 | 0.963 | **0.982** |
| DBPedia | Wikipedia | 0.513 | 0.513 | **0.533** | 0.521 | 0.525 |
| SCIDOCS | scientific | 0.144 | 0.151 | **0.163** | 0.108 | 0.136 |
| FEVER | Wikipedia | **0.931** | 0.911 | 0.929 | 0.907 | 0.913 |
| Climate-FEVER | Wikipedia | 0.442 | 0.433 | **0.444** | 0.438 | 0.383 |
| SciFact | scientific | 0.597 | 0.579 | **0.696** | 0.316 | 0.410 |

## First-Stage Retrieval

Setup: search over all documents

In-domain Results: suboptimal

| Dataset | Comparison Systems: BM25 | SBERT* | TAS-B* | SPLADEv2* | Ours: Neural PG-RANK with SBERT | with TAS-B |
|---|---|---|---|---|---|---|
| MS MARCO dev | 0.228 | **0.434** | 0.407 | 0.433 | 0.416 | 0.401 |