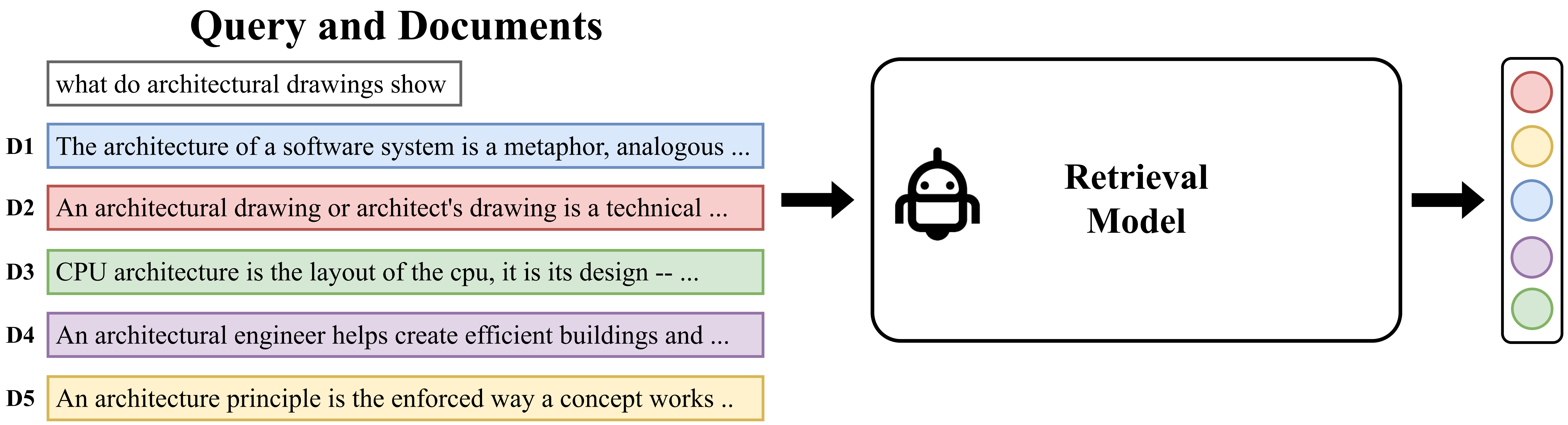
Policy-Gradient Training of Language Models for Ranking

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

Cornell Bowers CIS Computer Science | CORNELL TECH

Background

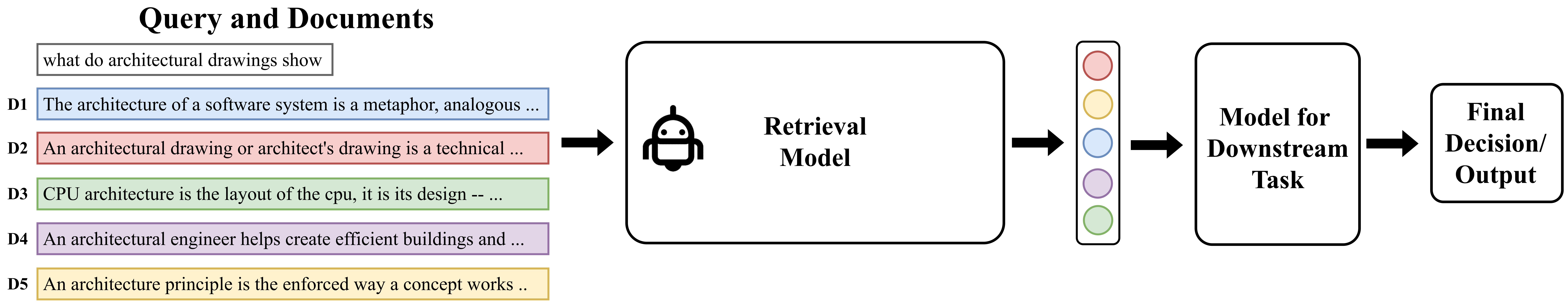
- <u>Task definition of retrieval:</u> rank documents based on their relevance to a query



Background

- Task definition of retrieval: rank documents based on their relevance to a query

- Application of retrieval models: ranked documents are input to some downstream models; separate from training retrieval models



Background

- <u>Conventional training objectives:</u> contrastive loss, requiring ground truth annotation for relevant documents and estimation for truly irrelevant documents (i.e. hard negatives)

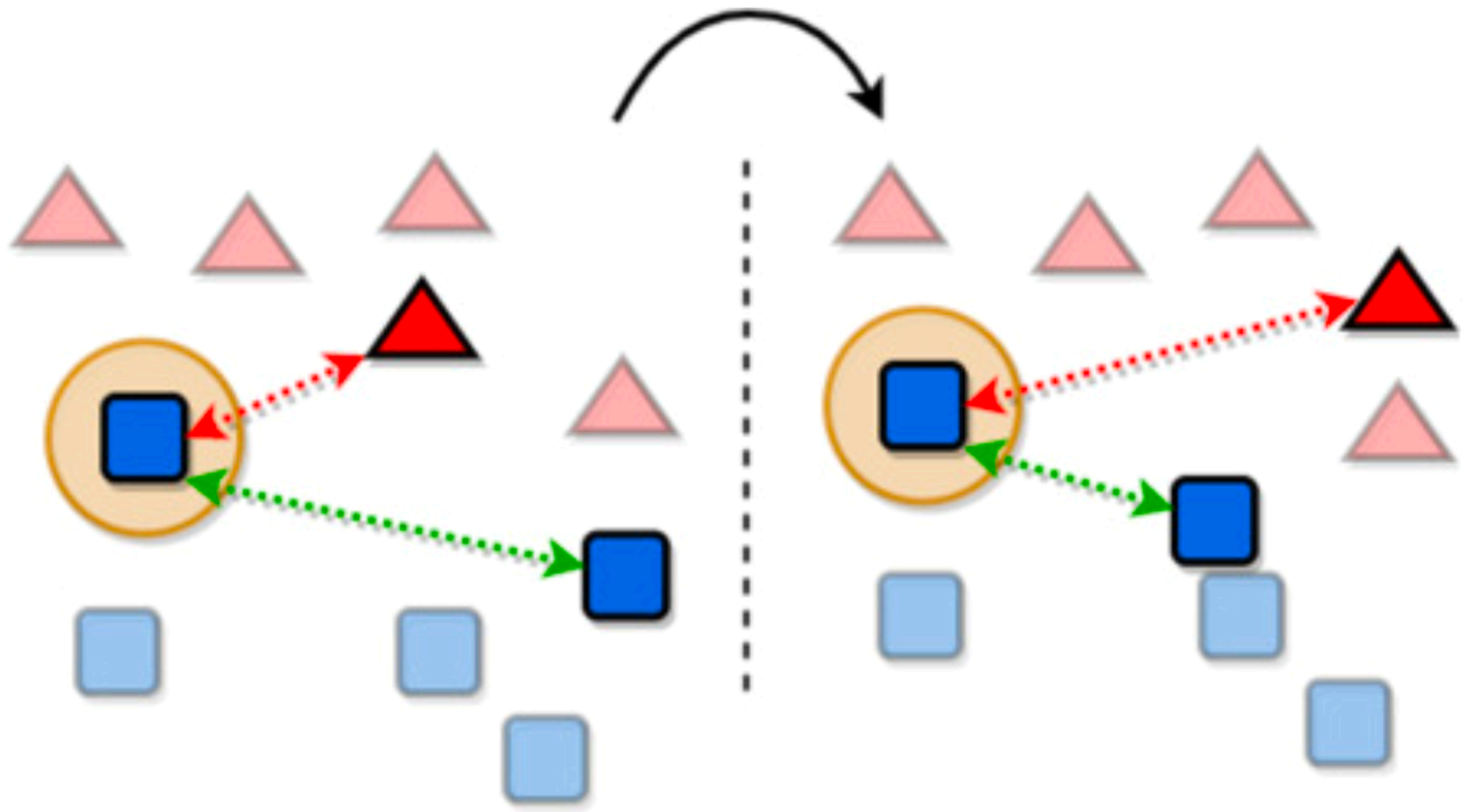


Image from https://www.sciencedirect.com/topics/computer-science/contrastive-loss

Overview

- We introduce **Neural PG-RANK** to train LLM-based retrieval models that directly optimize downstream decision-making quality

Overview

- We introduce **Neural PG-RANK** to train LLM-based retrieval models that directly optimize downstream decision-making quality
  - Learns to rank by instantiating a LLM as a <u>Plackett-Luce ranking policy</u>
  - <u>End-to-end training</u> of retrieval models as part of larger pipelines via <u>policy gradient</u>
  - Can optimize the ranker for <u>any cardinal loss function</u> evaluating the downstream decisions

Overview

- We introduce **Neural PG-RANK** to train LLM-based retrieval models that directly optimize downstream decision-making quality

Setting

- Define the utility of a ranking policy for a given query

Utility function

$$U(\pi|q) = \mathbb{E}_{r \sim \pi(\cdot|q)} \left[\Delta(r|q) \right]$$

Query

Ranking

Setting

- Define the utility of a ranking policy for a given query

Utility function

$$U(\pi|q) = \mathbb{E}_{r\sim\pi(\cdot|q)}\left[\Delta(r|q)\right]$$

Query

Ranking

- Learning objective is to learn a ranking policy that optimizes the expected utility over the query distribution

$$\pi^\star = \operatorname*{argmax}_{\pi\in\Pi} \mathbb{E}_{q\sim\mathcal{Q}}\left[U(\pi|q)\right]$$

Method

- Define a Plackett-Luce ranking policy

Definition 1 (Plackett-Luce Model (Plackett, 1975; Luce, 1959)). *Given the utility scores of the N items, $\boldsymbol{w} = [w_1, w_2, \cdots, w_N]^T$, the probability of observing a certain ordered list of these items, $(i_1, i_2, \cdots, i_N)$, is defined as*

$$p((i_1, i_2, \cdots, i_N); \boldsymbol{w}) = \prod_{j=1}^{N} \frac{\exp(w_{i_j})}{\sum_{l=j}^{N} \exp(w_{i_l})}. \tag{1}$$

Method

- Define a Plackett-Luce ranking policy
  - expressed as a product of softmax distributions
  - based on query-document relevance scores

Scoring function

Query

Doc

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

Method

- We use REINFORCE

$$\begin{aligned}\nabla_\theta U(\pi_\theta|q) &= \nabla_\theta \mathbb{E}_{r\sim\pi_\theta(\cdot|q)}\left[\Delta(r|q)\right] \\ &= \mathbb{E}_{r\sim\pi_\theta(\cdot|q)}\left[\nabla_\theta \log \pi_\theta(r|q)\Delta(r|q)\right]\end{aligned}$$

Method

- We use REINFORCE

  + Monte Carlo sampling with N samples

  + Variance reduction with leave-one-out baseline

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

Method

- We use REINFORCE

  + Monte Carlo sampling with N samples

  + Variance reduction with leave-one-out baseline

  + nDCG@10 as utility function

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

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

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

Utility score for the partial ranking til k

Utility

- nDCG@10: score between 0 and 1; higher means better ranking
- nDCG@10 is an approximation of the downstream utility in our work
- Assumption: higher nDCG@10 relates to better downstream task performance

Experimental Setup

- <u>Data:</u> MS MARCO for training; BEIR [Thakur et al., 2021] for evaluation

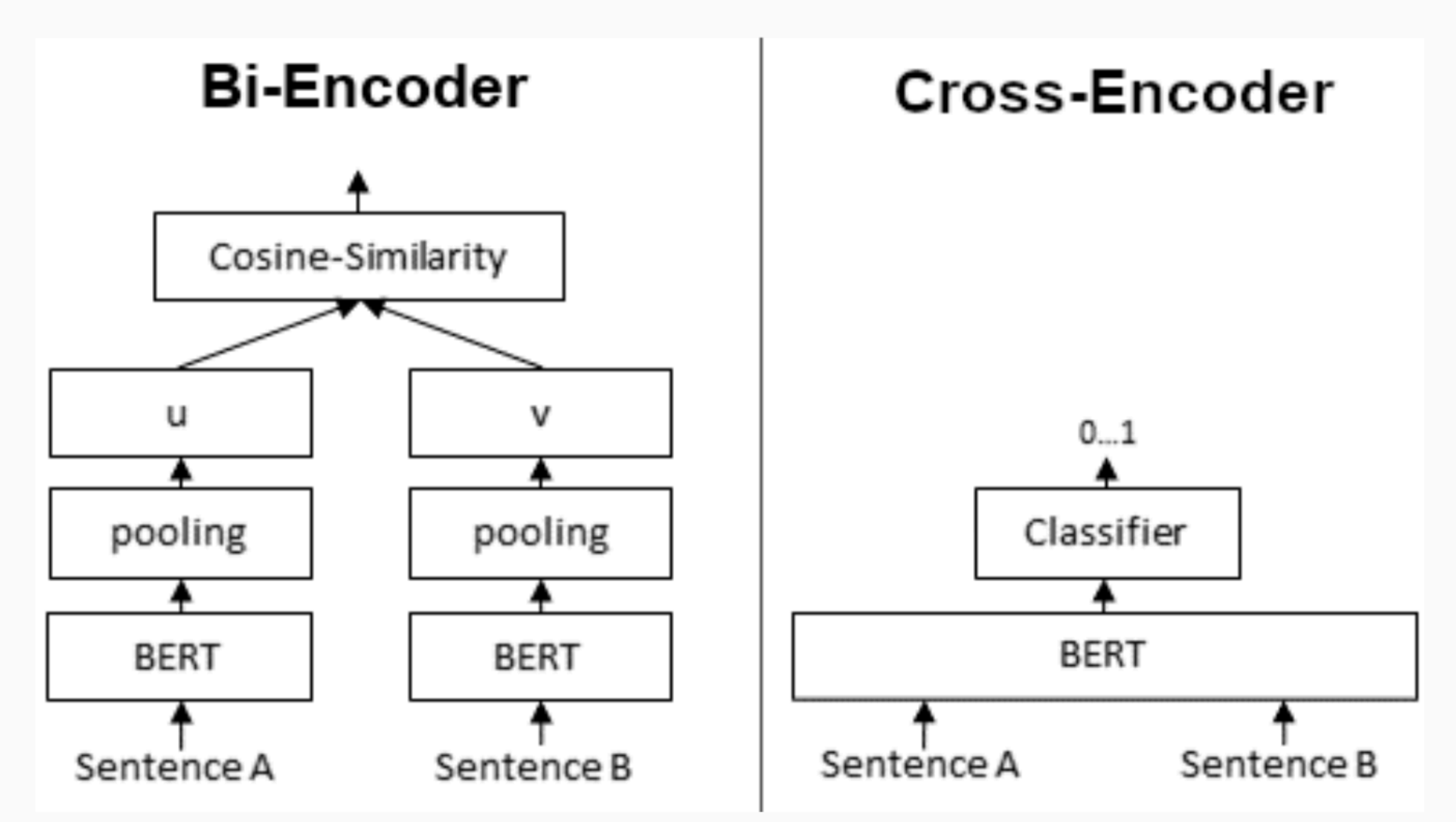
Split (→)					Train	Dev	Test			Avg. Word Lengths	
Task (↓)	**Domain (↓)**	**Dataset (↓)**	**Title**	**Relevancy**	**#Pairs**	**#Query**	**#Query**	**#Corpus**	**Avg. D / Q**	**Query**	**Document**
Passage-Retrieval	Misc.	MS MARCO [45]	✗	Binary	532,761	—-	6,980	8,841,823	1.1	5.96	55.98
Bio-Medical Information Retrieval (IR)	Bio-Medical	TREC-COVID [65]	✓	3-level	—-	—-	50	171,332	493.5	10.60	160.77
	Bio-Medical	NFCorpus [7]	✓	3-level	110,575	324	323	3,633	38.2	3.30	232.26
	Bio-Medical	BioASQ [61]	✓	Binary	32,916	—-	500	14,914,602	4.7	8.05	202.61
Question Answering (QA)	Wikipedia	NQ [34]	✓	Binary	132,803	—-	3,452	2,681,468	1.2	9.16	78.88
	Wikipedia	HotpotQA [76]	✓	Binary	170,000	5,447	7,405	5,233,329	2.0	17.61	46.30
	Finance	FiQA-2018 [44]	✗	Binary	14,166	500	648	57,638	2.6	10.77	132.32
Tweet-Retrieval	Twitter	Signal-1M (RT) [59]	✗	3-level	—-	—-	97	2,866,316	19.6	9.30	13.93
News Retrieval	News	TREC-NEWS [58]	✓	5-level	—-	—-	57	594,977	19.6	11.14	634.79
	News	Robust04 [64]	✗	3-level	—-	—-	249	528,155	69.9	15.27	466.40
Argument Retrieval	Misc.	ArguAna [67]	✓	Binary	—-	—-	1,406	8,674	1.0	192.98	166.80
	Misc.	Touché-2020 [6]	✓	3-level	—-	—-	49	382,545	19.0	6.55	292.37
Duplicate-Question Retrieval	StackEx.	CQADupStack [25]	✓	Binary	—-	—-	13,145	457,199	1.4	8.59	129.09
	Quora	Quora	✗	Binary	—-	5,000	10,000	522,931	1.6	9.53	11.44
Entity-Retrieval	Wikipedia	DBPedia [21]	✓	3-level	—-	67	400	4,635,922	38.2	5.39	49.68
Citation-Prediction	Scientific	SCIDOCS [9]	✓	Binary	—-	—-	1,000	25,657	4.9	9.38	176.19
Fact Checking	Wikipedia	FEVER [60]	✓	Binary	140,085	6,666	6,666	5,416,568	1.2	8.13	84.76
	Wikipedia	Climate-FEVER [14]	✓	Binary	—-	—-	1,535	5,416,593	3.0	20.13	84.76
	Scientific	SciFact [68]	✓	Binary	920	—-	300	5,183	1.1	12.37	213.63

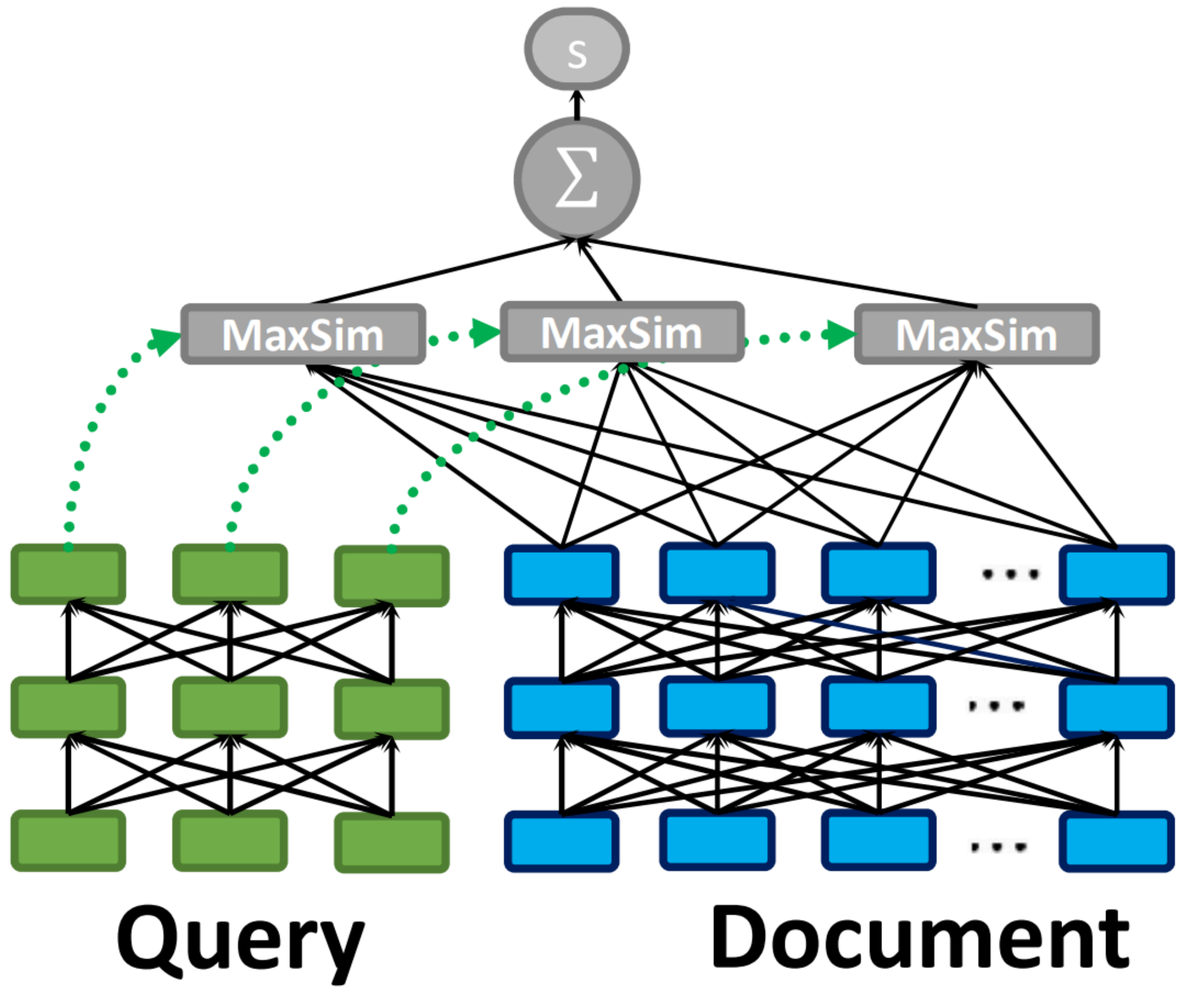
Experimental Setup

- Data: MS MARCO for training; BEIR [Thakur et al., 2021] for evaluation
- Evaluation metric: nDCG@10
- Our ranking policy: either SBERT [Reimers & Gurevych, 2019] or TAS-B [Hofstätter et al., 2021] as warmstart, with Neural PG-RANK method as fine-tuning

Experimental Setup

- Data: MS MARCO for training; BEIR [Thakur et al., 2021] for evaluation
- Evaluation metric: nDCG@10
- Our ranking policy: either SBERT [Reimers & Gurevych, 2019] or TAS-B [Hofstätter et al., 2021] as warmstart, with Neural PG-RANK method as fine-tuning





(d) Late Interaction

Image from https://www.sbert.net/examples/applications/cross-encoder/README.html and ColBERT

Experimental Setup

- Data: MS MARCO for training; BEIR [Thakur et al., 2021] for evaluation
- Evaluation metric: nDCG@10
- Our ranking policy: either SBERT [Reimers & Gurevych, 2019] or TAS-B [Hofstätter et al., 2021] as warmstart, with Neural PG-RANK method as fine-tuning
- Comparison systems: supervised learning SOTA bi-encoder models

Method	**Source of Negative Docs**			**Additional Supervision**	**Loss**
	In-Batch	BM25	Dense Model		
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

Result: Second-Stage Reranking

- <u>Setup:</u> search over a candidate set of 1k documents per query

Result: Second-Stage Reranking

- Setup: search over a candidate set of 1k documents per query
- In-domain results:
  - Performance gains with both warmstart models (nDCG@10)

Dataset	**Domain**	**Comparison Systems**			**Ours: Neural PG-RANK**	
		SBERT*	TAS-B*	SPLADEv2*	with SBERT	with TAS-B
MS MARCO dev	misc.	0.892	0.893	0.900	**0.987**	0.982

Result: Second-Stage Reranking

- Setup: search over a candidate set of 1k documents per query
- In-domain results:
  - Performance gains with both warmstart models (nDCG@10)

Dataset	Domain	Comparison Systems			Ours: Neural PG-RANK	
		SBERT*	TAS-B*	SPLADEv2*	with SBERT	with TAS-B
MS MARCO dev	misc.	0.892	0.893	0.900	**0.987**	0.982

  - More gains in terms of nDCG@k with smaller k (nDCG@1 below)

Dataset	Comparison Systems			Ours: Neural PG-RANK	
	SBERT*	TAS-B*	SPLADEv2*	with SBERT	with TAS-B
MS MARCO dev[‡]	0.826	0.819	0.830	**0.975**	0.965

Result: Second-Stage Reranking

- Out-of-domain results:

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

Result: Second-Stage Reranking

- <u>Out-of-domain results:</u>

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

Result: Second-Stage Reranking

- <u>Out-of-domain results:</u>

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**	<u>0.982</u>
TREC-DL 2019	misc.	<u>0.743</u>	**0.749**	**0.749**	0.742	0.741
TREC-COVID	bio-medical	**0.764**	0.711	<u>0.731</u>	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	<u>0.869</u>	**0.878**
HotpotQA	Wikipedia	0.747	0.785	0.834	**0.902**	<u>0.900</u>
FiQA-2018	finance	0.291	0.279	**0.342**	0.131	0.139
ArguAna	misc.	0.351	0.479	**0.480**	<u>0.354</u>	0.443
Touché-2020	misc.	**0.480**	0.423	<u>0.460</u>	0.363	0.361
Quora	Quora	0.962	**0.982**	<u>0.967</u>	0.963	**0.982**
DBPedia	Wikipedia	0.513	0.513	**0.533**	0.521	<u>0.525</u>
SCIDOCS	scientific	0.144	<u>0.151</u>	**0.163**	0.108	0.136
FEVER	Wikipedia	**0.931**	0.911	<u>0.929</u>	0.907	0.913
Climate-FEVER	Wikipedia	<u>0.442</u>	0.433	**0.444**	0.438	0.383
SciFact	scientific	<u>0.597</u>	0.579	**0.696**	0.316	0.410

Result: Second-Stage Reranking

- Setup: search over a candidate set of 1k documents per query
- In-domain results:
  - Performance gains with both warmstart models
  - More gains in terms of nDCG@k with smaller k (nDCG@1, 3, 5)
- Out-of-domain results:
  - Comparable generalization to in-domain results
  - Notable improvements on widely-studied QA datasets
  - Weaker in the domain of bio-medicine, science and finance

Result: First-Stage Retrieval

- Setup: search over all documents per query

Result: First-Stage Retrieval

- <u>Setup:</u> search over all documents per query
- <u>In-domain results:</u>
    - Suboptimal compared to warmstart models

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

Summary

- We introduce **Neural PG-RANK** to train LLM-based retrieval models that directly optimize downstream decision-making quality
    - 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
- When the training objective aligns with the evaluation setup, Neural PG-RANK yields remarkable in-domain performance improvement, with substantial out-of-domain generalization to some critical datasets employed in downstream QA tasks.