



Tricks for Sparse & Dense Retrieval Models

Breezedeus, 2021.12



IR: 检索 vs. 排序

- IR 系统流程: Retrievers → Rankers

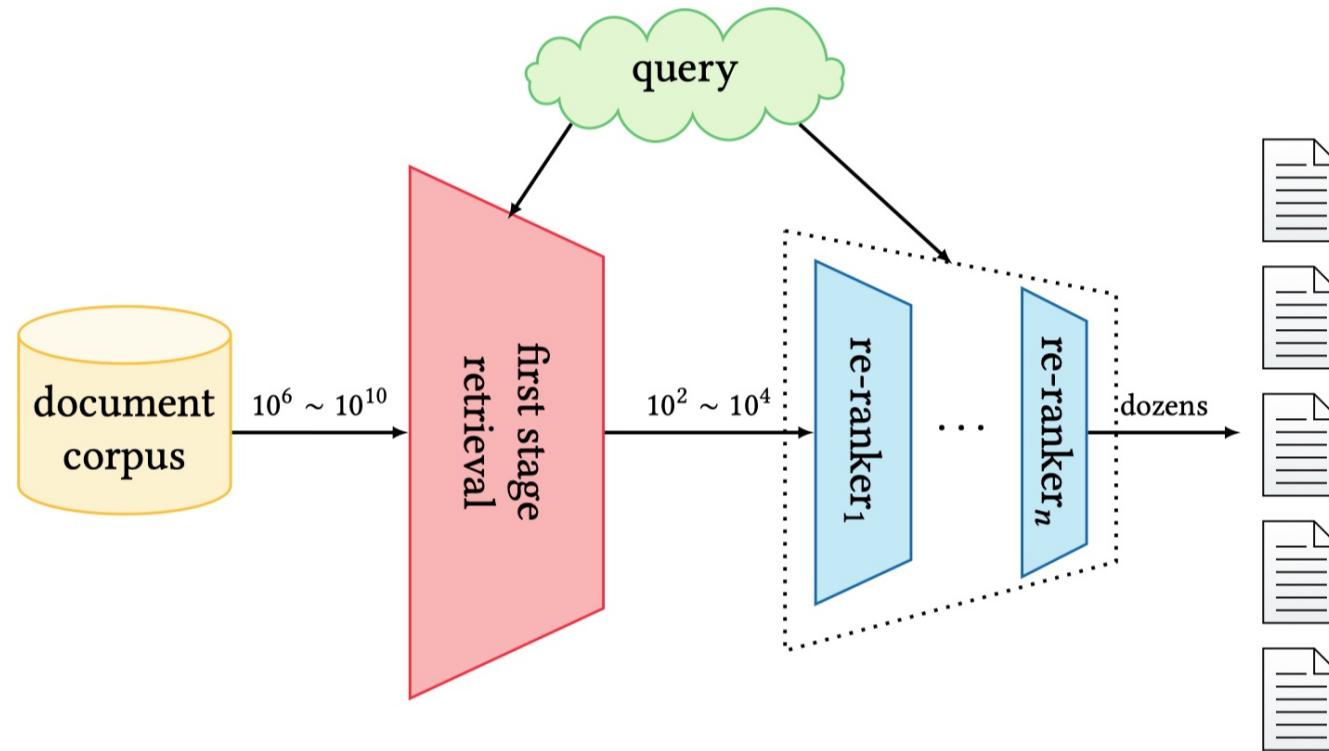


Fig. 1. The multi-stage architecture of modern information retrieval systems.

检索 vs. 排序

Retriever

- 处理文档多
- 要求计算快
- 召回覆盖全面

Ranker

- 处理文档少
- 要求排序准确，可容忍更高延时

Sparse vs. Dense

Sparse Retriever (SR)

代表模型：

- TF-IDF, BM25

优势：

- 无监督、领域迁移性好
- 效果不错

劣势：

- 基于字词，无法考虑语义
- 无法利用标注数据改进效果

Dense Retriever (DR)

代表模型：

- Siamese Network + BERT

优势：

- 考虑语义
- 可以通过训练获得优于SR的效果

劣势：

- 无监督训练的效果未必优于SR
- 领域迁移性不好



PART I: 如何优化 SR 效果

1. 通过训练获得词的权重值
2. 对 query 或 doc/passage 进行扩展
3.

通过训练获得词的权重值

DeepCT: Deep Contextualized Term Weighting framework

模型经过BERT编码后，利用MSE直接拟合每个token的权重值（重要性）：

$$\begin{aligned}\hat{y}_t &= \vec{w}T_t + b \\ \text{loss}_{MSE} &= \sum_t (y_t - \hat{y}_t)^2\end{aligned}$$

其中 T_t 就是BERT返回的token t 的向量。

DeepCT的关键是token目标权重 y_t 如何获得

DeepCT

对于Passage，作者建议使用 query term recall (QTR):

Target Term Weights for Training DeepCT. Proper target term weights should reflect whether a term is essential to the passage or not. We propose **query term recall** as an estimation of the ground truth passage term importance:

$$QTR(t, d) = \frac{|Q_{d,t}|}{|Q_d|}. \quad (3)$$

Q_d is the set of queries for which passage, $= d$ is judged relevant. $Q_{d,t}$ is the subset of Q_d that contains term t , and $QTR(t, d)$ is the query term recall weight for t in d . QTR is in the range of [0, 1].

这种计算方式需要较大的训练数据量，否则大部分值可能都是0

对于Query，作者建议使用类似的 term recall (TR):

Target Term Weights for Training DeepCT. Inspired by Zheng and Callan [36], DeepCT-Query uses **term recall** [35]:

$$TR(t, q) = \frac{|D_{q,t}|}{|D_q|}. \quad (5)$$

(3) D_q is the set of documents that are relevant to the query. $D_{q,t}$ is the subset of relevant documents that contains term t . Their ratio, $TR(t, q)$, is the term recall weight for term t in query q . Term recall is in the range of [0, 1]. Term recall is based on the assumption that

- HDCT 是把 DeepCT 推广到长文本 (doc) 场景

对 doc 进行扩展

doc2query

- 使用生成式模型Transformer， 输入为doc， 预测输出为这个doc可能回答的queries
- 然后把预测得到的queries追加到原始doc中， 追加后的新doc再利用BM25进行检索

docT5query

- docT5query 只是把doc2query中的生成模型替换为生成模型 T5， 会带来显著效果提升

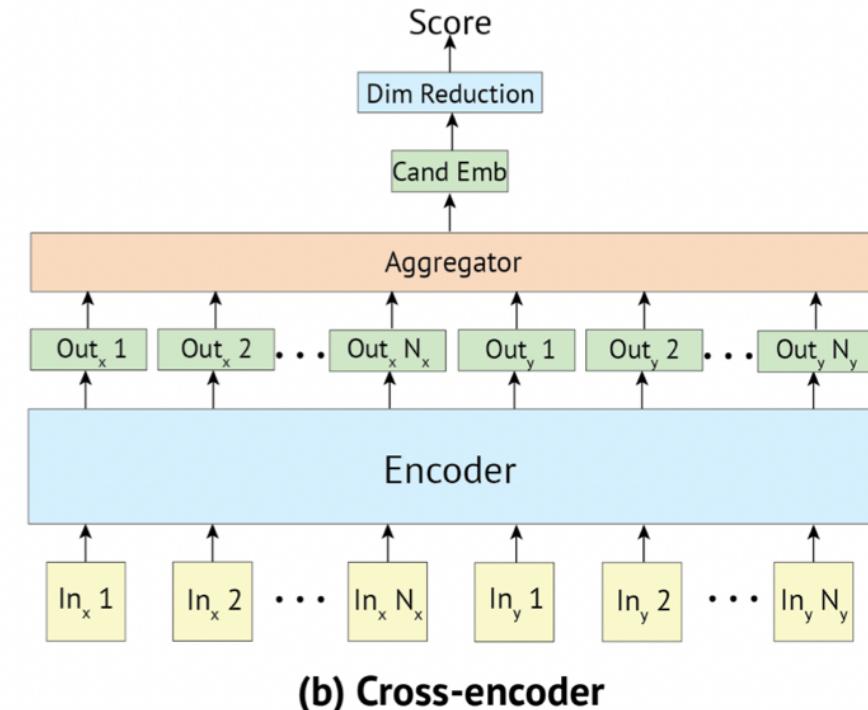
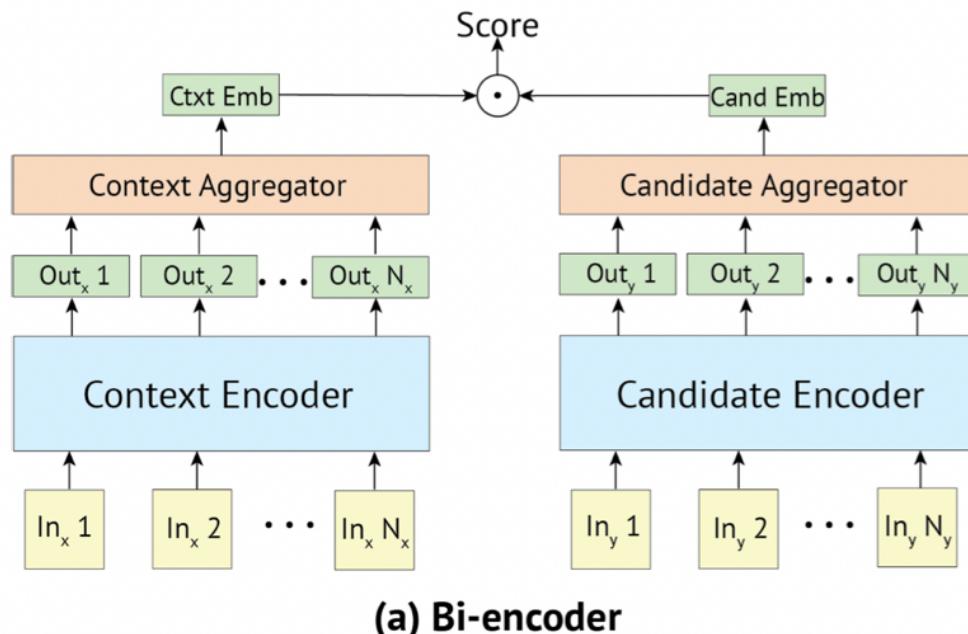


PART II: 如何优化 DR 效果

1. Complex Vector Representation
2. Hard Negative Sampling
3. Distillation of more expressive architectures
4. Hybrid of Sparse and Dense Retrievers
5. Improved Pretraining

BERT_{DOT} vs. BERT_{CAT}

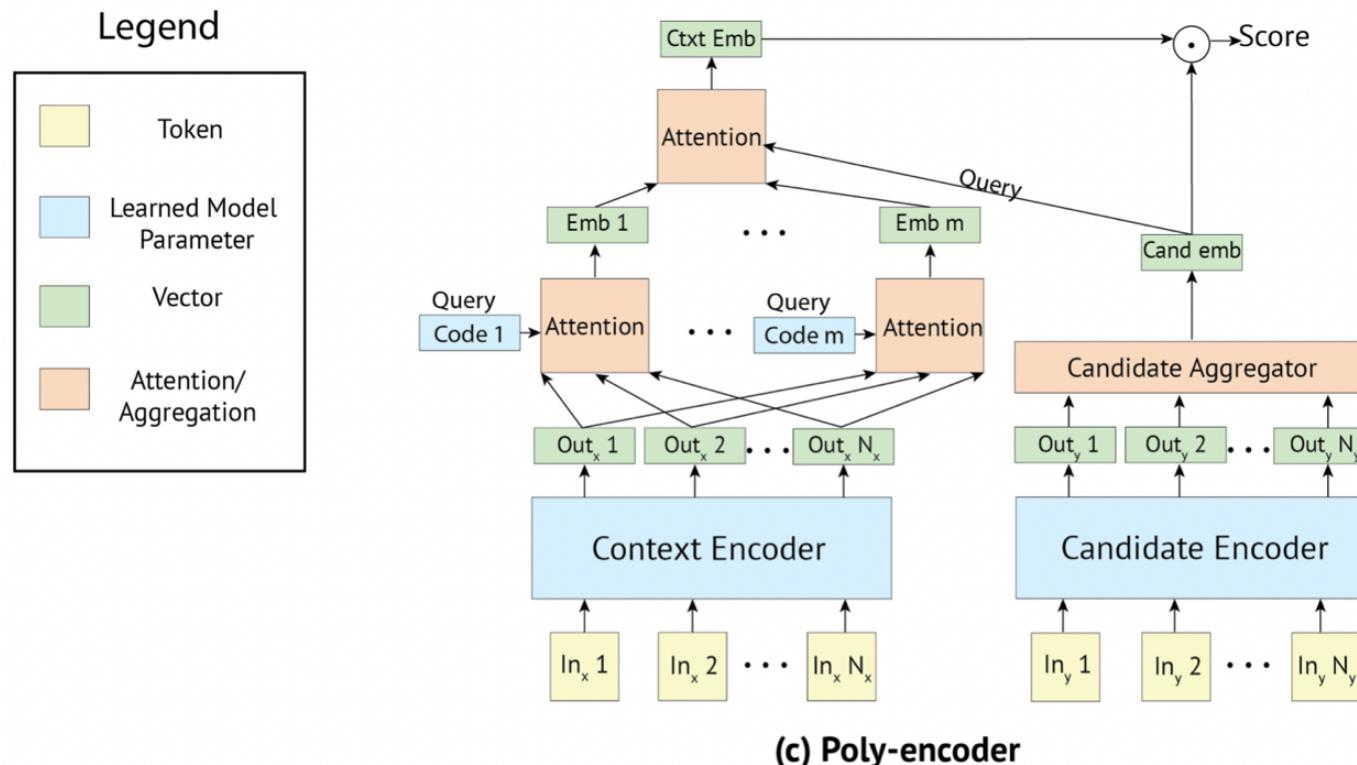
- BERT_{DOT}: 又名 Bi-encoder, Dual-encoder; Siamese 双塔结构
- BERT_{CAT}: 又名 Cross-encoder
- 通常效果上: BERT_{DOT} << BERT_{CAT}



Complex Vector Representation

Poly-encoder

- 使用多个向量来表示一个query/context, 原因是对话应用中context一般更长
- 训练方法: 类似 InfoNCE, 负样本来自同一个batch下随机选取的其他样本



Complex Vector Representation

ME-BERT: Multi-Vector Encoding from BERT

- 引入了多个向量来表示passage
- BERT会对每个token给出向量表示， ME-BERT 使用 passage中前 m 个 tokens 的向量（第一个token是 [CLS]），来作为此passage的表示向量组
- passage与query的相关性得分，是query的向量与这 m 个向量内积的最大值：

$$\max_{j=1\dots m} \langle f^q(\mathbf{x}), f_j^p(\mathbf{y}) \rangle$$

$m = 1$ 退化为 BERT_{DOT}。作者的实验中 $m = 3$ 或者 $m = 4$ 时效果最好

Complex Vector Representation

CoBERT

- 一般来说，query和passage的交互越往前，效果会越好
- CoBERT保留query和passage中每个词的表示向量，然后基于这些向量计算
MaxSim 相关性：

$$S_{q,d} = \sum_{i=1}^N \max_{j=1}^M Q_i \cdot D_j^T$$

其中 Q_i 和 D_j 分别对于query第 i 个词和passage第 j 个词的向量。

Complex Vector Representation

CoBERT V2

- CoBERT需要存储passage中每个词的向量，显然存储量消耗巨大
- CoBERT V2通过对词向量进行聚类，然后利用很少位数存储每个向量相对于中心的差异，以降低所需的存储量
- 同时，V2中也加入了蒸馏（老师模型：BERT_{CAT}，使用KL-Divergence loss），以及 **hard negative sampling + in-batch negatives** 来提升效果

Types of Negative Sampling

- **Random Negative Sampling**
 - 常使用 In-batch negatives 的方式获得随机负样本
- **Hard Negative Sampling**
 - **Static Hard Negatives**
 - 模型在训练过程中会很快拟合静态负样本，反而可能带来泛化性的降低
 - 单独使用static hard negatives的话，效果可能比random negatives还差
 - **Dynamic Hard Negatives**
 - Hard Negatives在训练过程中不断变化

Random Negative Sampling

QGen

- QGen 使用通用领域训练的query生成器在目标域生成 query, 之后使用 MultipleNegativesRanking (MNRL) 损失训练检索模型:

$$L_{\text{MNRL}}(\theta) = -\frac{1}{M} \sum_{i=0}^{M-1} \log \frac{\exp (\tau \cdot \sigma(f_{\theta}(Q_i), f_{\theta}(P_i)))}{\sum_{j=0}^{M-1} \exp (\tau \cdot \sigma(f_{\theta}(Q_i), f_{\theta}(P_j)))}$$

其中 Q 表示 query, P 表示 passage, M 表示 batch size, τ 控制 softmax 归一化的尖锐程度。

可见, QGen 就是query生成后, 使用 in-batch negatives 的 InfoNCE 进行训练

Hard Negative Sampling

- 随机采样获得的负样本绝大部分都无法为训练提供信息，过于简单
- 期望获得更有信息量的负样本
- 如何降低 **false negatives** (正样本被误识为负样本) 的影响?

Hard Negative Sampling

DPR: Dense Passage Retrieval

- 利用 $BERT_{DOT}$ 做检索，负样本不仅使用了 in-batch negatives，还利用BM25为每个样本选取一个得分最高的负样本
- 训练loss是InfoNCE
- 最新版中，DPR也加入了其他 hard negative sampling 方法，挖掘用的就是之前的 DPR模型。

Hard Negative Sampling

RocketQA, V2

- **Denoised Hard Negatives**: 依旧是使用已有的DR来检索出static hard negatives，但这其中可能存在一些false negatives，所以使用已训练好的 $BERT_{CAT}$ 模型做过滤，去掉 $BERT_{CAT}$ 预测置信度低的样本
- 类似GPL，使用 $BERT_{CAT}$ 模型对无标注数据打标签，做数据扩充。但只保留其中置信度高的正负样本。这些 $BERT_{CAT}$ 打标签后的正负样本加入到之前的训练样本中参与训练（相当于 **hard labeling**）
- V2 中把 denoised hard negatives 和 undenoised hard negatives 一起使用

Hard Negative Sampling

ANCE

- 论文中的场景是有正样本（知道每个query对应的一些相关正样本），但没有负样本
- 作者分析得出in-batch negatives大部分样本是不提供信息量的，对训练不起作用
- 如何选取负样本？直观上可以利用BM25来计算打分，然后把得分高又不在正样本中的样本作为负样本其实是让模型去拟合BM25模型的效果，容易把模型带偏

Hard Negative Sampling

ANCE

- ANCE使用最新存储的checkpoint文件对应的模型获得doc对应的向量，然后构建索引，利用ANN (approximate nearest neighbor)技术获取得分最高但不在正样本中的样本作为负样本
- ANN索引的更新是异步的，一旦有了新的checkpoint文件后就重新计算doc向量然后更新索引。初始的负样本可以利用BM25获得
- Dynamic Hard Negative Sampling

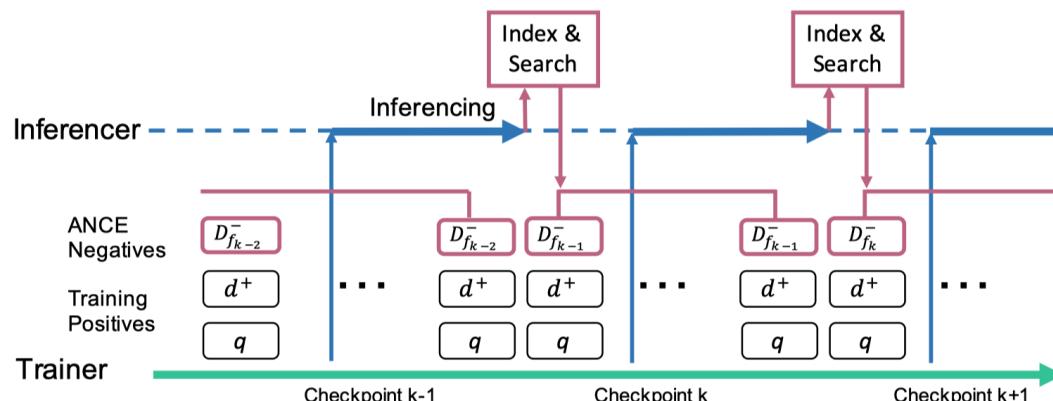


Figure 2: ANCE Asynchronous Training. The Trainer learns the representation using negatives from the ANN index. The Inferencer uses a recent checkpoint to update the representation of documents in the corpus and once finished, refreshes the ANN index with most up-to-date encodings.

Hard Negative Sampling

STAR: Stable Training Algorithm for dense Retrieval

- STAR 在训练前利用已有模型获得static hard negatives，这些样本在训练过程中不改变；然后使用 in-batch negatives作为随机负样本；训练时组合使用这两种负样本：

$$\theta^* = \alpha \cdot L_{\text{random}}(\theta) + L_{\text{static}}(\theta) \quad (0 < \alpha \ll 1)$$

- Loss 使用 RankNetLoss：

$$\mathcal{L}(d^+, d^-) = \log \left(1 + e^{f(q, d^-) - f(q, d^+)} \right)$$

- Static Hard + Random Negative Sampling

Hard Negative Sampling

ADORE: Algorithm for Directly Optimizing Ranking pErformance

- 固定doc的embedding向量，只训练query对应的embedding
- 在每次迭代前利用最新的模型更新query的表示向量，然后重新选取它们各自的hard negatives (类似ANCE)
- Dynamic Hard Negative Sampling
- Loss 使用的是 LambdaLoss：

$$\mathcal{L}(d^+, d^-) = \Delta\mathcal{M} \cdot \log \left(1 + e^{f(q, d^-) - f(q, d^+)} \right)$$

where $\Delta\mathcal{M}$ is the change in target IR metric when swapping the ranking positions of d^+ and d^-

STAR 和 ADORE 可以组合使用。先用STAR训练doc 向量，再用ADORE进一步训练 query 向量

Better Negative Sampling ?

TAS: Topic Aware Sampling

- 在训练之前，先利用 k -means 算法将 query 聚类到 k 个类别（如 $k = 2000$ ）中：

$$\arg \min_C \sum_{i=1}^k \sum_{q \in C_i} \|q - v_i\|^2$$

其中 query 的向量 q 由基线模型获得，如 BERT_{DOT}。 v_i 为 C_i 类别的中心

- 聚类后同类别的query可以认为是比较相像的。在构建 batch 的时候，我们可以先随机抽样 n 个类别，然后在每个类别中随机抽样 $\lfloor b/n \rfloor$ 个 query
- 这种batch抽样方式称为 Topic Aware Sampling (TAS)
 - 在后续的实验中，作者为 40 万个 query 聚类出 $k = 2000$ 个类别，并设 batch size 大小为 $b = 32$ ，组建 batch 时随机抽样的类别数为 $n = 1$ ，这样，每个 batch 中的样本都来自于同一个聚类类别，显然模型要对它们的正负样本作区分更困难

Better Negative Sampling ?

TAS-B: TAS-Balanced

- TAS只是选取出batch中的queries，对于pairwise的老师模型，如上面的 BERT_{CAT} 模型，还需要选取与各个query对应的正负样本，它们共同组成 (q, p^+, p^-)
- 可以在 TAS 的基础上进一步均衡 batch 内正负样本对的 margin 分布以减少 high margin (low information) 的正负样本对
 - 具体来说，针对每个 query，首先计算它对应的样本对集合的最小 margin 和最大 margin，然后将该区间分割为 $h = 10$ 个子区间，每个区间大小为 m
 - 为 query 配置 (p^+, p^-) 时，首先从这 h 个子区间中随机选择一个子区间 i ，然后从 margin 落在该子区间的正负样本对集合中随机采样一组 (p^+, p^-) 形成一个训练样本：

$$H(P_q, i) = \{(p^+, p^-) \mid m_{\min} + i \times m \leq M_t(q, p^+) - M_t(q, p^-) < m_{\min} + (i + 1) \times m\}$$

Better Negative Sampling ?

TAS-B

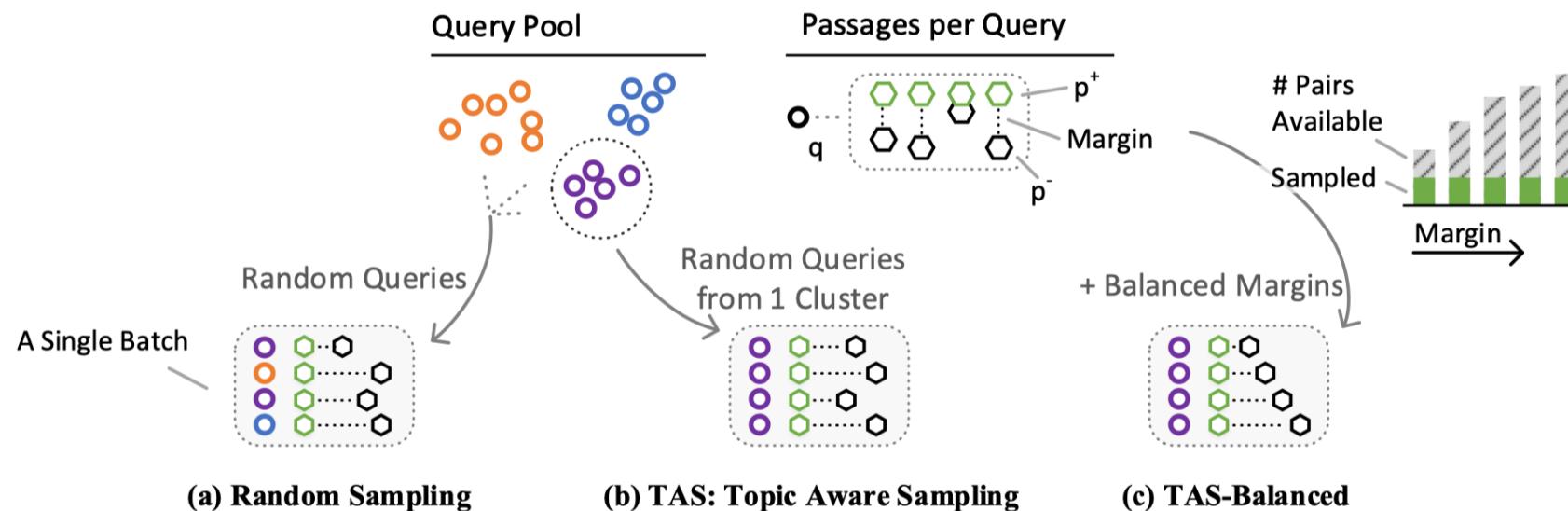


Figure 2: Comparison of batch sampling strategies. Each strategy has access to a pool of (clustered) queries; where each query has a set of relevant and non-relevant passage pairs with BERT_{CAT} score margins.

Distillation

| 蒸馏方法普遍有效！

- 可以有单个Teacher，也可以有多个Teachers，只要Teacher效果更好
 - BERT_{CAT}
 - ColBERT
- 蒸馏 Loss
 - i. KL-Divergence loss
 - ii. embedding MSE loss
 - iii. Margin-MSE loss

Distillation

TAS-B

两个Teacher模型

- 第一个 is **BERT_{CAT}** 模型
- 使用的是正负样本对形式的loss，作者使用了 **Margin-MSE loss**：

$$\mathcal{L}_{\text{Pair}}(Q, P^+, P^-) = \text{MSE}(M_s(Q, P^+) - M_s(Q, P^-), \\ M_t(Q, P^+) - M_t(Q, P^-))$$

其中 M_s 和 M_t 分别对应学生和老师模型

Margin-MSE loss 希望保留正负样本对之间的得分差异，而不只是保留顺序关系

Distillation

TAS-B

两个Teacher模型

- 第二个是 ColBERT。对于这个老师，使用了 **in-batch negative** 方法，把同一个batch中的所有其他样本都当做负样本：

$$\begin{aligned}\mathcal{L}_{\text{InB}}(Q, P^+, P^-) = & \frac{1}{2|Q|} \left(\sum_i^{|Q|} \sum_{p^-}^{P^-} \mathcal{L}_{\text{Pair}}(Q_i, P_i^+, p^-) \right. \\ & \left. + \sum_i^{|Q|} \sum_{p^+}^{P^+} \mathcal{L}_{\text{Pair}}(Q_i, P_i^+, p^+) \right)\end{aligned}$$

其中的 $\mathcal{L}_{\text{Pair}}$ 同前，即 Margin-MSE loss

- 最终的loss是上面两个loss之和，其中并未显式用到有监督数据：

$$\mathcal{L}_{DS}(Q, P^+, P^-) = \mathcal{L}_{\text{Pair}}(Q, P^+, P^-) + \mathcal{L}_{\text{InB}}(Q, P^+, P^-) \times \alpha$$

Distillation

GPL: Generative Pseudo Labeling

GPL 步骤如下：

1. 使用已存在的T5生成模型，对每个passage生成 3 个queries，作为正样本
2. 对每个query，使用已存在的DR模型检索出 50 个hard negatives
3. 使用已存在的 BERT_{CAT} 模型，对每个 (q, p^+) 和 (q, p^-) 进行打分；然后使用 **MarginLoss Distillation** 训练当前模型
 - 生成模型生成的query未必靠谱，期望通过 BERT_{CAT} 模型的打分发现这些 false positives

GPL = QGen + Static Hard Negative Sampling + MarginLoss Distillation

Distillation

GPL: Generative Pseudo Labeling

GPL = QGen + Static Hard Negative Sampling + MarginLoss Distillation

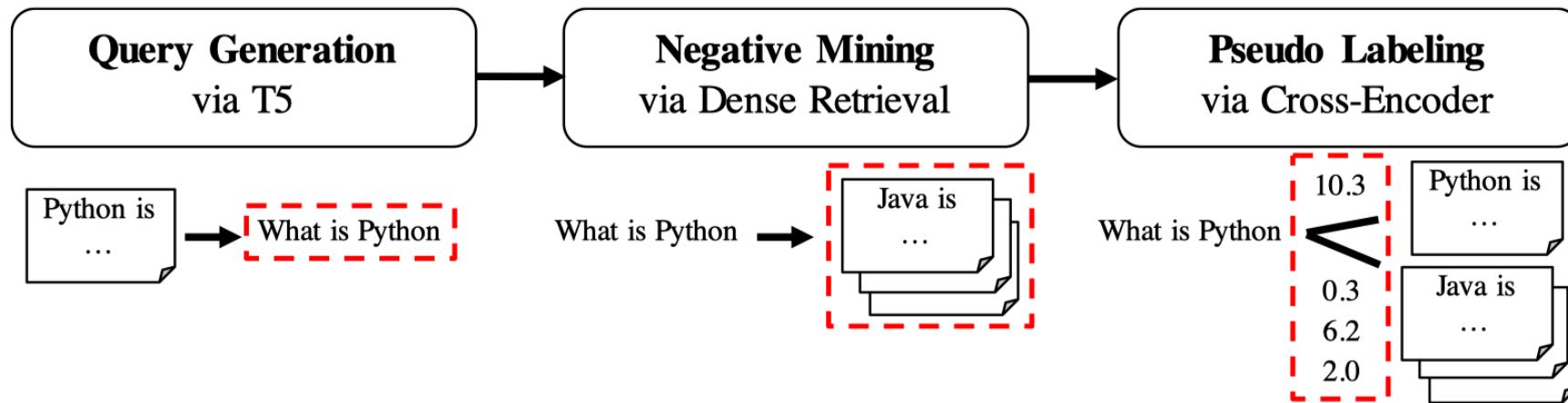


Figure 1: Generative Pseudo Labeling (GPL) for training domain-adapted dense retriever. First, synthetic queries are generated for each passage from the target corpus. Then, the generated queries are used for mining negative passages. Finally, the query-passage pairs are labeled by a cross-encoder and used to train the domain-adapted dense retriever. The output at each step is marked with dashed boxes.

Distillation

RocketQA V2

- **Dynamic Listwise Distillation**: 同时训练 ranker BERT_{CAT} 和 retriever BERT_{DOT} 模型，使用 ranker 模型来指导/蒸馏 retriever
- 对于 query q ，以及它对应的候选 passages $\mathcal{P}_q = \{p_{q,i}\}_{1 \leq i \leq m}$ ，retriever 的打分记为 $s_{\text{de}}(q, p)$ (Dual-encoder)，ranker 的打分记为 $s_{\text{ce}}(q, p)$ (Cross-encoder)。把它们分布化，然后计算它们的KL 距离，可以视为蒸馏中的KL loss

Distillation

RocketQA V2

$$\tilde{s}_{\text{de}}(q, p) = \frac{e^{s_{\text{de}}(q, p)}}{\sum_{p' \in \mathcal{P}_q} e^{s_{\text{de}}(q, p')}},$$

$$\tilde{s}_{\text{ce}}(q, p) = \frac{e^{s_{\text{ce}}(q, p)}}{\sum_{p' \in \mathcal{P}_q} e^{s_{\text{ce}}(q, p')}},$$

$$\mathcal{L}_{\text{KL}} = \sum_{q \in \mathcal{Q}, p \in \mathcal{P}_q} \tilde{s}_{\text{de}}(q, p) \cdot \log \frac{\tilde{s}_{\text{de}}(q, p)}{\tilde{s}_{\text{ce}}(q, p)}$$

loss中还包括一个 CE loss, 指导 ranker 训练:

$$\mathcal{L}_{\text{sup}} = -\frac{1}{N} \sum_{q \in \mathcal{Q}, p^+} \log \frac{e^{s_{\text{ce}}(q, p^+)}}{e^{s_{\text{ce}}(q, p^+)} + \sum_{p^-} e^{s_{\text{ce}}(q, p^-)}}$$

其中 N 为总样本数量, p^+ 和 p^- 为 \mathcal{P}_q 中的正负样本。

总 loss:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{sup}}$$

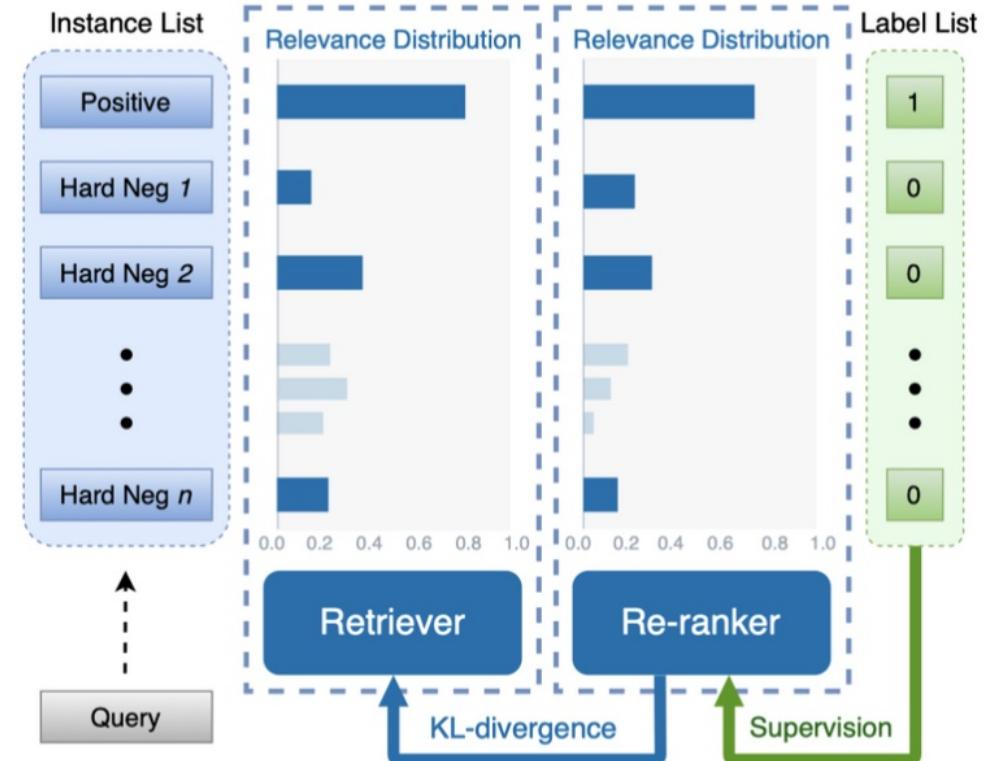


Figure 1: The illustration of dynamic listwise distillation in our approach.

Hybrid of Sparse and Dense Retrievers

- 融合 Dense 和 Sparse Retrievers 的得分，在一些任务上可进一步提升检索效果

$$s_{\text{hybrid}}(q, p) = s_{\text{Dense}}(q, p) + \alpha s_{\text{BM25}}(q, p)$$

其中 α 为超参数

- 合并时的一个小技巧（融合后模型返回 $\text{top-}k$ 检索结果）：

We adopt the normalization technique from Ma et al. (2021): If a passage p is found in the top- k' (with $k' > k$) of a dense retriever but not of BM25, then $\text{BM25}(q, p)$ is set to the minimum value from the top- k' results of BM25 (and vice versa).

Improved Pretraining

- 期望预训练好的模型，能够更好地 **zero-shot** 迁移到 target domain 检索任务上
- 思路
 - 使用无监督预训练，继续训练
 - 使用无监督对比学习，提高迁移性

Improved Pretraining

ICT (Inverse Cloze Task) 预训练任务

Inverse Cloze Task (ICT) Given a passage p consisting of n sentences, $p = \{s_1, \dots, s_n\}$, the query q is a sentence randomly drawn from the passage, $q = s_i, i \sim [1, n]$, and the document d is the rest of sentences, $d = \{s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n\}$. See (q_1, d) in Figure 2 as an example. This task captures the semantic context of a sentence and was originally proposed by Lee et al. (2019).

Improved Pretraining

Spider 无监督对比学习

- 定义 **cross-passage recurring spans (RS)** 为一篇文档中在多个passage出现过的 n-grams 词组
- 选取其中一个RS所在附近的词句作为query，并随机决定是否要删掉此query中包含的RS
- 包含此RS的其他passage作为正样本，同文档其他passage作为负样本。训练使用 InfoNCE，最后的预测范式同 BERT_{DOT}

Improved Pretraining

Spider 无监督对比学习

- 如下图中 q'_1 未删除RS，而 q'_2 则删除了RS



Figure 2: An example of our pretraining approach: Given a document \mathcal{D} (e.g. the article "**Aaron**" in Wikipedia), we take two passages that contain a recurring span S . One of them is transformed into a short query (left) q' using a random window surrounding S , in which S is either kept (top) or removed (bottom). The second passage is then considered the target for retrieval p^+ , while a random passage from \mathcal{D} that does not contain S is considered the negative p^- (right). Each batch is comprised of multiple such examples, and the pretraining task is to find the positive passage p_i^+ for each query q'_i (solid line) from the passages of all examples (dashed lines).

Improved Pretraining

Contriever 无监督对比学习

- 使用 **independent cropping** 方法构建query和正样本，其实就是随机在一篇文档中选一个片段作为query，然后同文档中再随机选一个片段作为正样本
 - 所以query和正样本之间可能存在overlap
- 训练的方式同SimCLR和MoCo，同batch其他样本作为负样本
- 作者发现SimCLR和MoCo效果差不多，但MoCo可以不使用大batch，所以论文中使用的是MoCo的训练方式

中文数据集上的效果 from Baidu

模型	Recall@1	Recall@5	Recall@10	Recall@20	Recall@50
有监督训练 Baseline	30.077	43.513	48.633	53.448	59.632
有监督训练 In-batch Negatives	51.301	65.309	69.878	73.996	78.881
无监督训练 SimCSE	42.374	57.505	62.641	67.09	72.331
无监督 + 有监督训练 SimCSE + In-batch Negatives	55.976	71.849	76.363	80.49	84.809
Domain-adaptive Pretraining + SimCSE	51.031	66.648	71.338	75.676	80.144
Domain-adaptive Pretraining + SimCSE + In-batch Negatives	58.248	75.099	79.813	83.801	87.733

Thanks

