0.1 Gecko

谷歌 DeepMind 发布 Gecko：专攻检索，与大 7 倍模型相抗衡

Gecko: Versatile Text Embeddings Distilled from Large Language Models

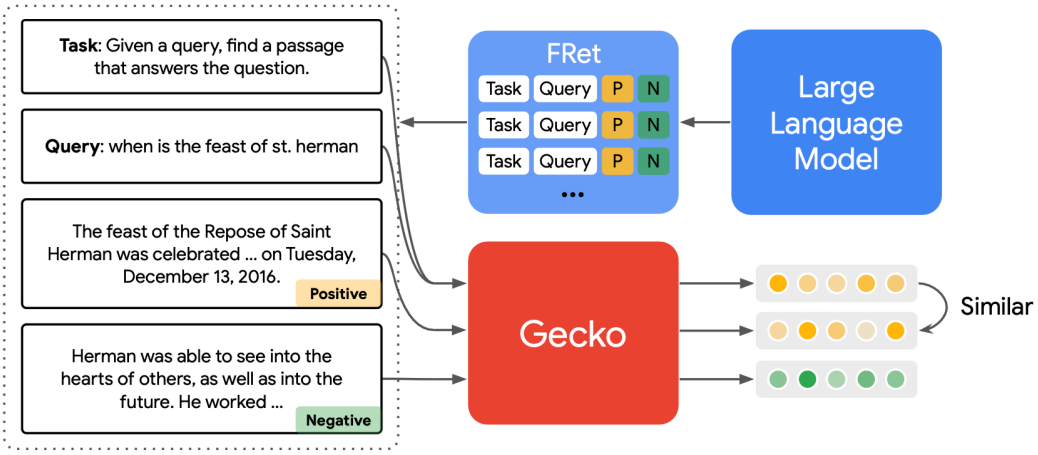

Figure 1 | Overview of Gecko. Gecko is a versatile text embedding model trained on a variety of tasks including document retrieval, semantic similarity, and classification. To train Gecko, we utilize FRet where queries are generated from LLMs, and their positive and negative passages are mined by LLMs.

主要包括两阶段：

- pre-finetuning：类似Large dual encoders are generalizable retrievers，自监督
- finetuning：2-step llm distillation，提出了一个 FRet 数据集（Few-shot Prompted Retrieval dataset）

0.1.1 Pre-finetuning

2 个数据集：

- 大型社区 qa 数据集：Large dual encoders are generalizable retrievers中的来自网上论坛和 qa 网站的问答 pair 对
- 去各种网站上爬了 title-body 的 pair 对，因为Text Embeddings by Weakly-Supervised Contrastive Pre-training发现这种自然出现的 pair 对对于 pre-finetuning embedding 模型很有用

pre-finetuning 的目标是让模型看到大量的多样性语料，对于一个预训练的语言模型 $\mathcal{M}$，长度为 n 的句子对应的 contextual 向量 $\mathbf{W} \in \mathbb{R}^{n \times d}$，对于任务特征 t，数据集 $\mathcal{D}_{\text{pre}} = \{(q_i, p_i)\}_{i=1}^{N}$，得到的向量如下：

$$\begin{aligned}\mathbf{q}_i &= \text{mean\_pool}_{|t|+|q_i|}\left[\mathcal{M}\left(t \oplus q_i\right) \in \mathbb{R}^{(|t|+|q_i|)\times d}\right] \in \mathbb{R}^d \\ \mathbf{p}_i &= \text{mean\_pool}_{|p_i|}\left[\mathcal{M}\left(p_i\right) \in \mathbb{R}^{|p_i|\times d}\right] \in \mathbb{R}^d\end{aligned}$$

对于 batchsize 为 B 的样本来说，inbatch 负例，$\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$，其 loss 如下：

$$\mathcal{L}_{\text{pre}} = \frac{1}{B}\sum_{i=1}^{B}\left[-\log \frac{e^{\text{sim}(\mathbf{q}_i, \mathbf{p}_i)/\tau}}{\sum_{j=1}^{B} e^{\text{sim}(\mathbf{q}_i, \mathbf{p}_j)/\tau}}\right]$$

在这个阶段没有用 hard 负例，用了能适配设备的最大 batchsize，这是Towards General Text Embeddings with Multi-stage Contrastive Learning和Text embeddings by weakly-supervised contrastive pre-training里的经验。

0.1.2 FRet

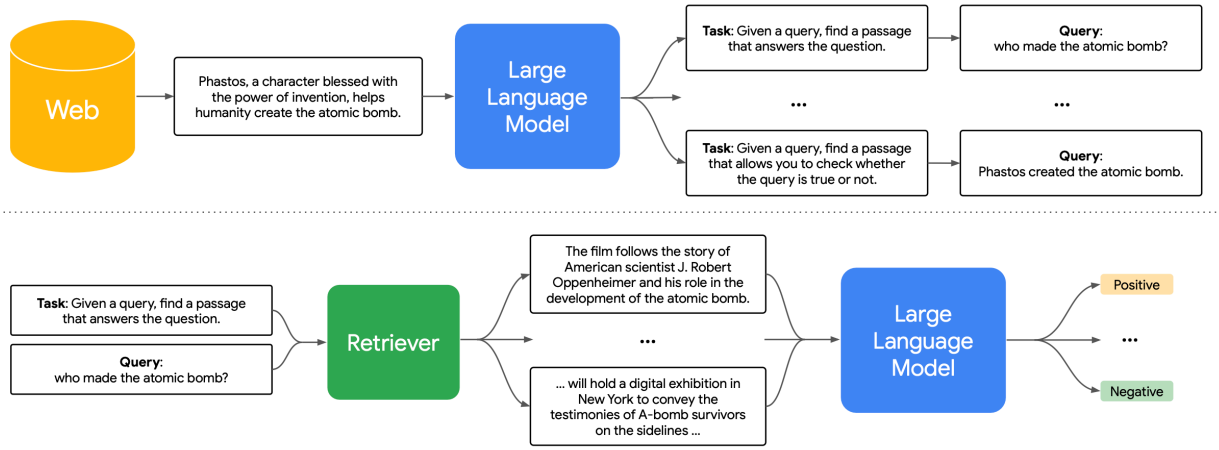

Figure 2 | Overview of FRet. Given a sampled passage from the web, FRet first utilizes LLMs to generate a relevant task and a query for the passage (top). Then, each query and task is fed into a pre-trained embedding model to obtain nearest neighbor passages, which are then scored by the LLM to mine positive and negative passages (bottom). Note that the original web passage does not necessarily become a positive passage as LLMs can find a more relevant passage as shown above.

0.1.2.1 LLM-based Diverse Query Generation

从 web 语料 C 中抽出一个段落 p_{seed}，$\mathbb{P}_{\text{QG}}$ 是一个固定的 few-shot 的 prompt，让 LLM 生成任务描述和这个 task 的 query

$$\text{LLM}\left(\mathbb{P}_{\text{QG}}, p_{\text{seed}}\right) \rightarrow (t, q)$$

生成的任务 t 例如：

- question answering: `Given a query, find a passage that has the answer to the query`
- fact checking: `Given a query, find a passage that allows you to check whether the query is true or not`

为了保证多样性：

- 网页库本身就有很多的 topic 和很多的写作风格
- 在 prompt 里加多样的任务描述，来让模型生成的 query 保证多样性。

0.1.2.2 LLM-based Positive and Negative Mining

对于某个 query q，之前的工作一般会直接把输入的段落 p_{seed} 当成正样本，但实践中发现 p_{seed} 一般比较长，而生成的 query 一般只关注其中一小部分，所以可能在整个语料库中有比 p_{seed} 更准确的答案。因此，通过如下方法构造了一个 FRet 数据集：

- 先把 p_{seed} 当成正样本，in-batch 负例训一个 embedding 模型。
- 用这个模型从文档库中检索出 top N 的相似段落 $P = \left\{p^{(1)}, \ldots, p^{(N)}\right\}$
- 用生成 query 的 LLM 给这 N 个文档排序，有两种排序方法：
  - query likelihood：参考Improving passage retrieval with zero-shot question generation，给定段落 p，衡量 query q 的 likelihood，$\text{QL}(q, p) = \text{LLM}\left(q \mid p, \mathbb{P}_{\text{QL}}\right)$，其中的 prompt 参考PaRaDe: Passage Ranking using Demonstrations with Large Language Models包括了判断 query likelihood 的指令，以及一些相关 query+ 段落的 few-shot。基本思想是：**如果 q 和 p 高度相关，那么从 p 生成 q 的概率应该很高。**
  - relevance classification：参考Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels，给定 query q 和段落 p 后，衡量特定相关性 label 的 log likelihood：

$\mathrm{RC}(q,p) = \mathrm{LLM}(\text{ label } \mid q, p, \mathbb{P}_{\mathrm{RC}})$。基本思想是：**让 LLM 直接判断 q 和 p 的相关程度，并输出一个相关性标签。**

- 通过标准的 Reciprocal Rank Fusion（RRF，倒数融合，Reciprocal rank fusion outperforms condorcet and individual rank learning methods）方法得到 rank 函数，对上面两个方法的排序结果 merge：$R(q,p) = 1/r_{\mathrm{QL}}(q,p) + 1/r_{\mathrm{RC}}(q,p)$ …其实就是 rankindex
- 基于得分筛选样本：
  - 正样本：$p^+ = \arg\max_{p \in P} R(q,p) = p_1$
  - hard 负例：排名第 N 位的样本 p_N，也可以从除了第 1 个外的 N-1 个里随机 sample

0.1.3 Unified Fine-tuning Mixture

除了 FRet，还融合了多个公开数据集：

- Natural Questions：Natural Questions: A Benchmark for Question Answering Research
- HotpotQA：Hotpotqa: A dataset for diverse, explainable multi-hop question answering
- FEVER：Fever: a large-scale dataset for fact extraction and verification
- MedMCQA：Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering
- SNLI：A large annotated corpus for learning natural language inference
- MNLI：A broad-coverage challenge corpus for sentence understanding through inference
- MIRACL：多语言的数据集Miracl: A multilingual retrieval dataset covering 18 diverse languages
- huggingface 上的一些分类数据集

将这些数据集处理成一个统一的格式，发现不同的任务对 prompt 的格式敏感程度不同，，非对称任务（如 BEIR）对格式更敏感，而对称任务的性能相对稳定。

- 对称格式（Symmetric Formatting）：输入和目标使用相同的格式。
  - 输入：task: {task} | query: {input}
  - 目标：task: {task} | query: {target}
- 非对称格式（Asymmetric Formatting）：输入和目标使用不同的格式。
  - 输入：task: {task} | query: {input}
  - 目标：title: {title} | text: {target}

参考One Embedder, Any Task: Instruction-Finetuned Text Embeddings，对于每个文本 x，找到另一个同样 label 是 y 的样本当成正样本 x^+，随机找一个 label 不是 y 的当作负样本 x^-。

实践中，在一个 batch 中，同一个 x^+ 可能出现 overlap，会造成 in-batch negatives 中的 false negative 问题。参考poe 的回复，即同一个 mini-batch 中假设有 (x1, x1+, x1-) 和 (x2, x2+, x2-)，可能出现 x1+ 与 x2 或 x2+ 相同或非常相似的情况。那么：

- 对于 x1 来说，x1+ 是正例。
- 但是如果 x1+ 与 x2 相同或非常相似，模型可能会错误地认为 x1+ 应该与 x2 区分开来，因为 x2 是另一个样本的输入。
- 或者如果 x1+ 与 x2+ 相同或非常相似，模型可能会混淆应该如何处理这个重叠的样本。

因为在理想情况下，x1+ 应该只是 x1 的正例，而不应该被视为任何其他样本的负例。但由于重叠，模型可能错误地将 x1+ 视为 x2 或其他样本的负例。解决方法，给每个三元组分配唯一的 id，让模型专注于在给定 x 的情况下，区分 x+ 和 x-。

有 M 个数据集 $\left[\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(M)}\right]$，每个数据集是 $\mathcal{D}^{(m)} = \left\{(t_i, q_i, p_i^+, p_i^-)\right\}_{i=1}^{N}$，$t$ 是任务描述，给定一个 batch_size=B 的 batch，同 batch 里的其他 query 可以看成SamToNe: Improving Contrastive Loss for Dual Encoder Retrieval Models with Same Tower Negatives中说的 same-tower negative（对同模态的效果比较好，例如这边都是 query）：

$$\mathcal{L}_{\text{main}} = \frac{1}{B}\sum_{i=1}^{B}\left[-\log \frac{e^{\text{sim}(\mathbf{q}_i \mathbf{p}_i^+)/\tau}}{\sum_{j=1}^{B}\left(e^{\text{sim}(\mathbf{q}_i \mathbf{p}_j^+)/\tau} + \mathbb{1}_{[j\neq i]} e^{\text{sim}(\mathbf{q}_i, \mathbf{q}_j)/\tau}\right) + e^{\text{sim}(\mathbf{q}_i \mathbf{P}_i^-)/\tau}}\right]$$

同时还参考Matryoshka representation learning的俄罗斯套娃加了一个 MRL 的 loss，让模型适配不同的 dim，gecko 是 768 和 256 的 dim