2020最后一篇！就是这么"硬"！召回系统就该这么做！

原创 十方 炼丹笔记 2020-12-31

收录于话题
#必读论文系列 16 #搜索推荐前沿算法 65

↑↑↑关注后"星标"**炼丹笔记**

炼丹笔记“硬”货

作者：十方，一品炼丹师

不知道多少人还记得《是"塔"!是"塔"!就是"它"，我们的双塔!》那篇，那篇介绍了国内外各个大厂做召回的用的双塔模型，其中提到一篇《Embeding-based Retrieval in FaceBook Search》，还跟大家强烈建议，该篇必读，不知道有多少炼丹师认真读了？什么？你还没读！没关系，十方今天就给大家解读这篇论文。

很多炼丹师往往迷恋于各种复杂的网络结构，比如某市值跌了几个“百”的大厂，每年都有各种花里胡哨的论文，这些结构有用吗？既然能发论文肯定有用（手动滑稽）。为什么十方在众多论文中强推"脸书"这篇呢？先给大家看下脸书的"双塔"。

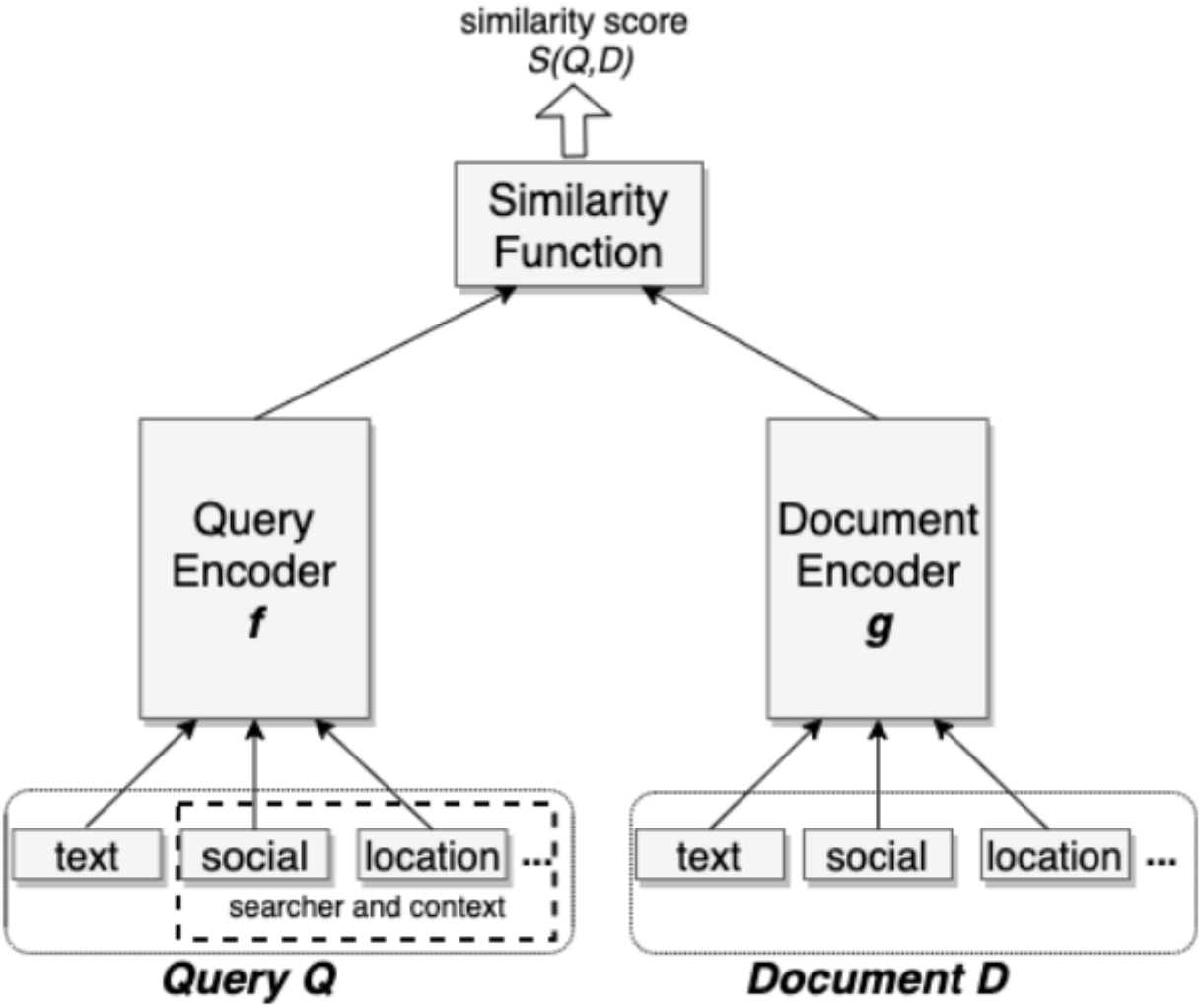

Figure 2: Unified Embedding Model Architecture

徐丹笔记

看完结构后，会不会有点劝退，什么！十方你就给我看这个，普普通通的双塔？Attention呢？Bert呢？FM呢？RNN呢？没错，这篇论文的精髓，不是网络结构，而是你在做召回时会遇到的方方面面的问题，以及解决方案，十方给大家慢慢揭晓。

对于一个搜索引擎而言，往往由两个层构成，一个叫召回层，另一个叫排序层。召回层的目的就是在低延时，低资源利用的情况下，召回相关的documents。排序层就是通过很复杂的算法(网络结构)把和query最相关的document排序到前面。论文的题目，简单直白的告诉了大家，用embeding 表示query和document来做召回。

论文提到，召回的难点，主要体现在候选集合非常庞大，处理亿级别的documents都是正常操作。不同于面部识别召回，搜索引擎的召回需要合并**字面召回**和**向量召回**两种结果。"脸书"的召回，还有其他难点，"人"的特征，在"脸书"的搜索尤其重要。

先膜拜下"脸书"的召回系统:

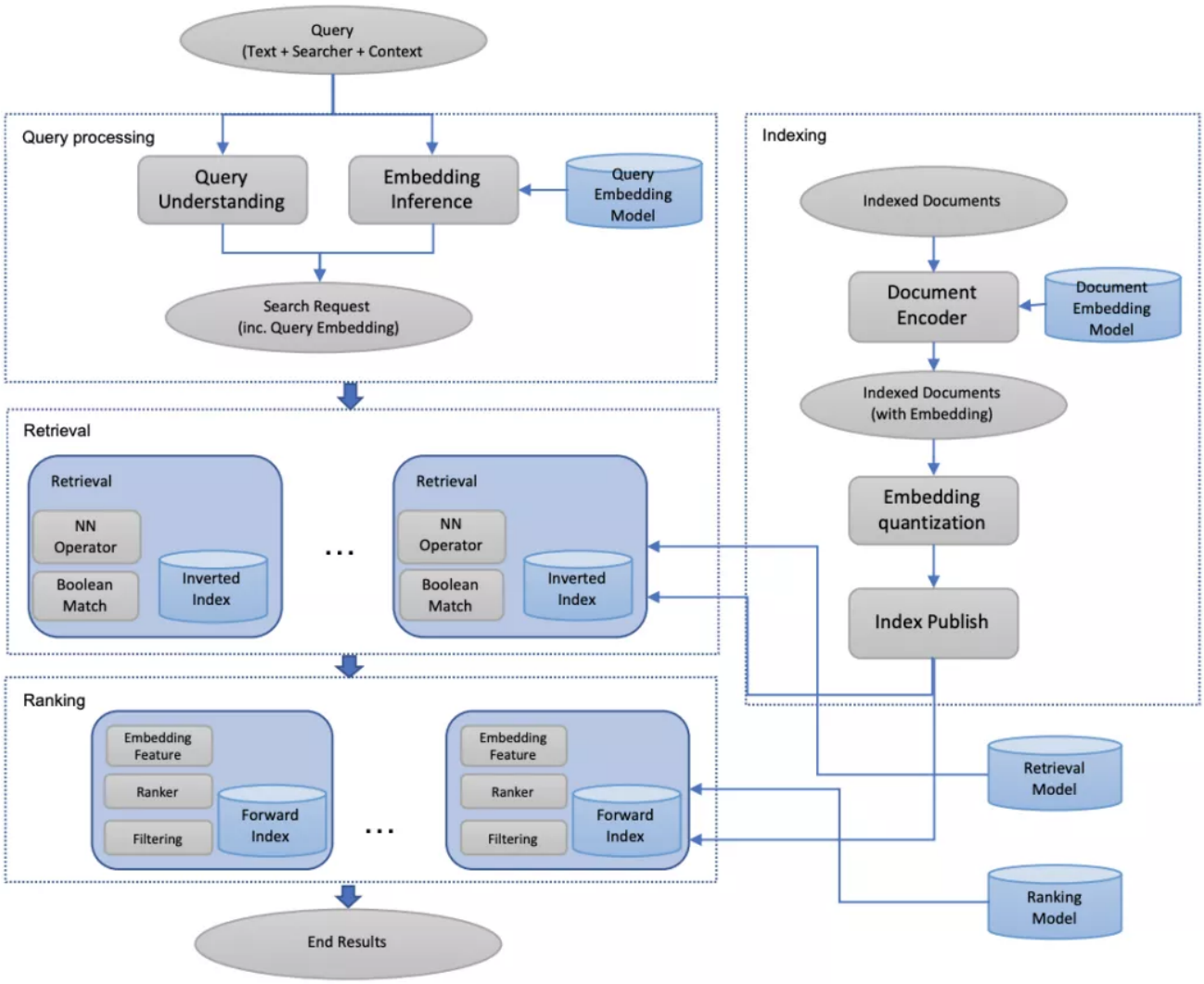

Figure 1: Embedding Based Retrieval System Overview

我们可以清楚的看到几个大模块。

- Query处理
- 索引模块
- 召回模块
- 排序模块

关于召回模型

既然是召回模型，那评价指标，当然要有recall，我们看下"脸书"的召回指标。给定一个Query，和目标结果集合T={t1,t2,...,tN}还有模型召回的top K {d1,d2,...,dk}，recall@K定义如下：

$$recall@K = \frac{\sum_{i=1}^{K} d_i \in T}{N}$$

至于Target集合怎么来的，它是有真实用户点击和专家标注组成的。模型训练用的是双塔配合triplet loss。

$$L = \sum_{i=1}^{N} \max(0, D(q^{(i)}, d_+^{(i)}) - D(q^{(i)}, d_-^{(i)}) + m)$$

炼丹笔记

该loss中的D，就是1-cos(emb_q, emb_d)。

论文提到很重要的一点是，调节m值影响最终召回5%-10%，炼丹师看到这，是不是又要把自己triplet loss的模型回炉重造了。然后"脸书"作者，又语出惊人了。如果给每个正例采n个负例，候选集合大小为N，模型实际优化的top K 召回率，K约等于N/n。

关于训练数据

训练数据的构造，应该是做召回最重要且没有之一的事了。花大精力去探索一两个不常见的特征，增加个什么attention结构，都没有构造一个良好的训练数据带来的增益要高。对于正样本用有点击的即可，负样本构造方式就有很多了。

- 随机负样本：对于每个query，在候选集合中随机的去选。
- 未点击的曝光: 对于每个query，随机的采样有曝光未点击的样本。

论文提到，用未点击的曝光作为负样本训练出来的模型非常糟糕，十方在实践中也发现了这个问题。原因是这部分负样本太hard了，这么hard当然要放到精排去学，召回任务最重要的是快速把和query相关的documents拉出来。如果召回阶段就能把曝光未点击的过滤掉，那还要精排干嘛。

论文对正样本还做了一些有趣的探索：

- 点击样本：这个不难理解，用户有点击行为，是因为最终曝光的结果是符合用户当前意图的(误点除外)。
- 曝光样本：作者认为召回就是粗粒度的排序，因此召回阶段就是要召回排序会打高分的documents，既然样本会曝光，说明排序模型认为这些样本分高，因此召回阶段应把这些样本当作正样本，不管有没有点击。

论文提到，实验中无论使用点击样本作为正样本，还是曝光样本作为正样本，效果最终都差不多。两个样本一起用，也没有质的提高。为了提高模型表现，必须要加入些hard的负样本。

关于召回的hard样本挖掘，"脸书"表示现有的大部分研究，都是针对图像领域的，在召回领域，没有明确"类别"的概念，于是论文作者给了自己的解决方案。

hard负样本挖掘:论文提到，他们发现top K召回结果大部分是同文本的，也就是说模型并没有充分利用社交特征。这主要因为随机负样本对于模型而言，因为和query文本完全不同，模型太容易学偏，认为文本一

样就是需要召回的。为了能使模型对相似的结果能有所区分，所以我们可以找到那些embeding很近，但实际上是负样本，让模型去学。一种方法是在线hard负样本挖掘，这个思路就是in-batch负采样，在一个batch内，有n个相关的query和document，对于任意一个query，其他的document都是它的负样本，但是由于每个batch也是随机产生的，in-batch内负采样并不能获得足够的hard负样本。所以就有了离线hard负样本采样。论文提到，在实验中发现，简单用hard负样本，效果是比用随机负样本要差的，主要原因是hard负样本需要非文本的特征区分，而easy负样本主要用文本特征区分，因此需要调整采样策略。论文还提到一点，hard负样本取排序模型排在101-500效果最好(所以其实要用semi-hard的样本)，而且hard负样本需要和easy负样本混合在一起用。混合方法有两种。（1）easy:hard = 100 : 1 (2) 先用hard样本学习，再基于该模型用easy样本学习。这两种方法效果都很好。

关于特征工程

论文提到，"脸书"在融合了各种特征，也给召回带来了不同程度的提升。特征罗列如下：

- 文本特征："脸书"使用字符级的n-gram特征，与单词级别的n-gram相比，词典大小大大缩简，OOV问题(拼写错误或其他问题)也得到了解决，实验也证明字符级别的特征，泛化能力也很好。当然字符和单词混用，也带来了泛化性能的提升(+1.5% recall)。因为单词级词典较大，需要用hash的方法去处理。
- 位置特征：用户的城市，地域，国家，语言。
- 社交embeding特征："脸书"有丰富的社交图谱，基于图神经网络训练出用户实体的embedding，直接作为召回模型的输入。

下图是不同特征带来的增益:

Table 1: Group Embedding Improvement with Feature Engineering

Unified Embedding	Abs. Recall Gain
+ location features	+ 2.20%
+ social embedding features	+ 1.77%

炼丹笔记

关于ANN

这里十方看完挺意外的，脸书并没有用Annoy Tree，HNSW这些近邻检索方法，而是用了PQ算法(矢量量化编码算法，关键是码本的建立和码字搜索算法。比如常见的聚类算法，就是一种矢量量化方法。而在相似搜索中，向量量化方法又以PQ方法最为典型)，这个本文就不赘述了(大家真的特别感兴趣留言，十方单独写一篇)，感兴趣的可以看下Product quantization，论文链接：
https://lear.inrialpes.fr/pubs/2011/JDS11/jegou_searching_with_quantization.pdf

"脸书"用这种ANN serving，主要考虑到他们现有索引系统，用这种方法能最简单的融合向量召回。对于向量召回阶段，论文提到他们也做了大量的调参。有个很重要的一点，就是论文提到，当模型发生改变(比如增加了负样本)，ANN的参数也需要重调。

关于排序优化

我们都知道排序阶段的结果，会成为召回的训练样本，而排序的输入又是召回的输出，这样模型学的就是有偏的，次优的，因此论文提出两个解决办法。

- 排序模型加召回模型的embedding特征:这样ranker能学到召回模型学到的特征，也能学到一个大概的相似度特征。具体做法是把query和document的embeding cosine，Hadamard product，和embeddings直接作为排序阶段的特征，实验证明，用cosine效果最好。
- 人工标注样本:把ANN召回的结果落到日志中，再由人工去标注，最后再喂给召回模型训练，提升召唤准确率。

关于模型融合

论文提到两种模型融合方式：

加权融合：该方法其实就是简单的线性加权，比如我有n个模型，就有n个query embedding，n个document embedding，正常加权如下:

$$S_w(Q,D)=\sum_{i=1}^{n}\alpha_i\cos(V_{Q,i},U_{D,i})$$

这样加权，就要算n次cos了，而且ann也要做n次，召回的解也不是最优的，有没有办法把权重加到embedding里呢？当然是可以的，很容易推导出下式:

$$E_Q=(\alpha_1\frac{V_{Q,1}}{\|V_{Q,1}\|},\cdots,\alpha_n\frac{V_{Q,n}}{\|V_{Q,n}\|})$$

$$E_D=(\frac{U_{D,1}}{\|U_{D,1}\|},\cdots,\frac{U_{Q,n}}{\|U_{Q,n}\|}).$$

很容易证明这样修改embeding后:

$$\cos(E_Q,E_D)=\frac{S_w(Q,D)}{\sqrt{\sum_{i=1}^{n}\alpha_i^2}\cdot\sqrt{n}}.$$

因此我们只需要线上用通过所有模型预估结果计算出EQ，线下用ED建ANN索引即可。

级联融合:就像瀑布一样地先去召回一个比较大的集合，再用加入hard负样本的模型过滤掉一部分，再输出到后面的排序模型。这里特别提到hard负样本不要用曝光未点击，而是要用上文提到的离线hard负样本。

结论

这篇论文，看似朴实，但其实写了很多大家在做召回时会遇到的问题，并给出了合理的解决方案。该论文的很多经验都值得我们借鉴，避免我们花大量时间去踩坑。

炼丹笔记公众号