搜索中的Query扩展技术

原创 丁香园大数据NLP 丁香园大数据 2020-05-06

前言

最近，我司各条业务线对于搜索优化的需求日益增多，NLP组也将对搜索业务给予更多的工作支持。后续分享，我们会关注过往的知识图谱、短本文理解等相关技术如何落地到搜索业务中。

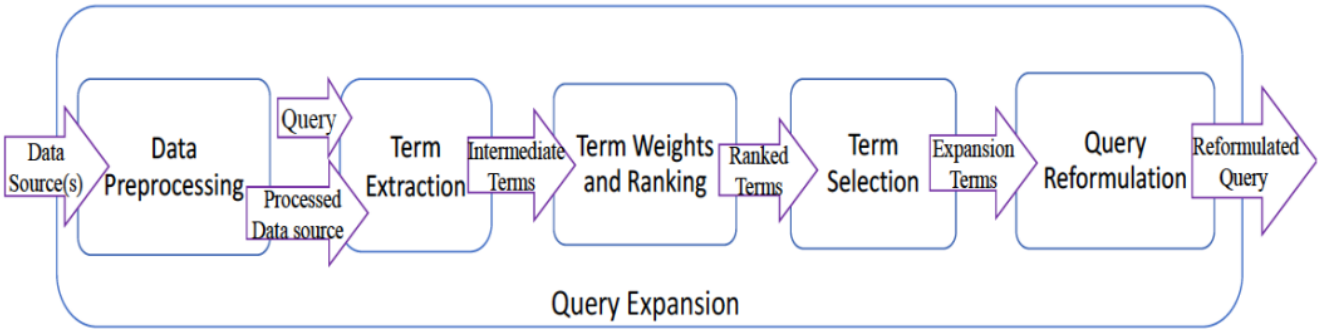
影响搜索结果的因素有很多，包括对**短文本的正确理解**（实体词识别、纠错、意图分析等）、**长文本良好结构化**（关键词抽取、主题词抽取、文本分类等）以及**排序模型**（召回策略、LTR、语义匹配等）。各种优化算法落在以上三个步骤中，对不同指标产生影响。

对于搜索优化，我们的建议是从**召回策略**开始着手。理由是这个步骤与实际业务方最近，当理清业务逻辑后，可以快速实施,看见变化。另外，**召回**阶段是整个搜索流程中的**基石**，所有后续的排序都基于召回的候选列表，先规划好召回策略，才可能尽量避免后续调整基石，导致与后续**“精排”**相互影响的境地。

那么改进**召回**我们一般会做些什么呢？首先一定是通过产品分析、用户调研来了解什么内容适合在这个搜索场景里展示，随后抡起大刀修改检索的字段或公式。有了baseline之后，我们在观察检索回的内容有什么问题。可能是没有匹配内容，可能是最匹配的内容排序靠后，或者可能是除了字面匹配，其他内容相关性差等等。此时，就可以上一些影响**召回**的模块，比如**Query词权重分配**、**动态时效性判定**、**Query扩展**等。后面会陆续有文章分享其他技术，本文我们先关注如何做**Query扩展**。

总的来说，**召回**于搜索是满足检索内容的大概范围，排序是次要的，需要关注的是Query与召回列表的相关度（字词层面和主题层面）。我们需要**Query扩展技术**的原因这里大致将它们归纳为**三**个方面。**首先**，用户输入的Query普遍较短，平均2-3个词，可能无法很好命中需要找的内容；**其次**，Query中的词通常会与多个主题关联，搜索引擎根据简短的几个词检索返回的内容可能不是用户所关心的那个主题；**另外**，用户可能对自己找的东西只有一个大致的概念（举个栗子，假设用户想要找“黑人抬棺”的视频，但是并不知道这个词的准确表述，转而使用“棺材”、“黑人” 这些词来搜索），Query扩展此时可理解为类似联想的功能，或者可以理解为将某个“罕见”搜索词改写成“常见”搜索词。

完整的**Query扩展**技术路线可见下图



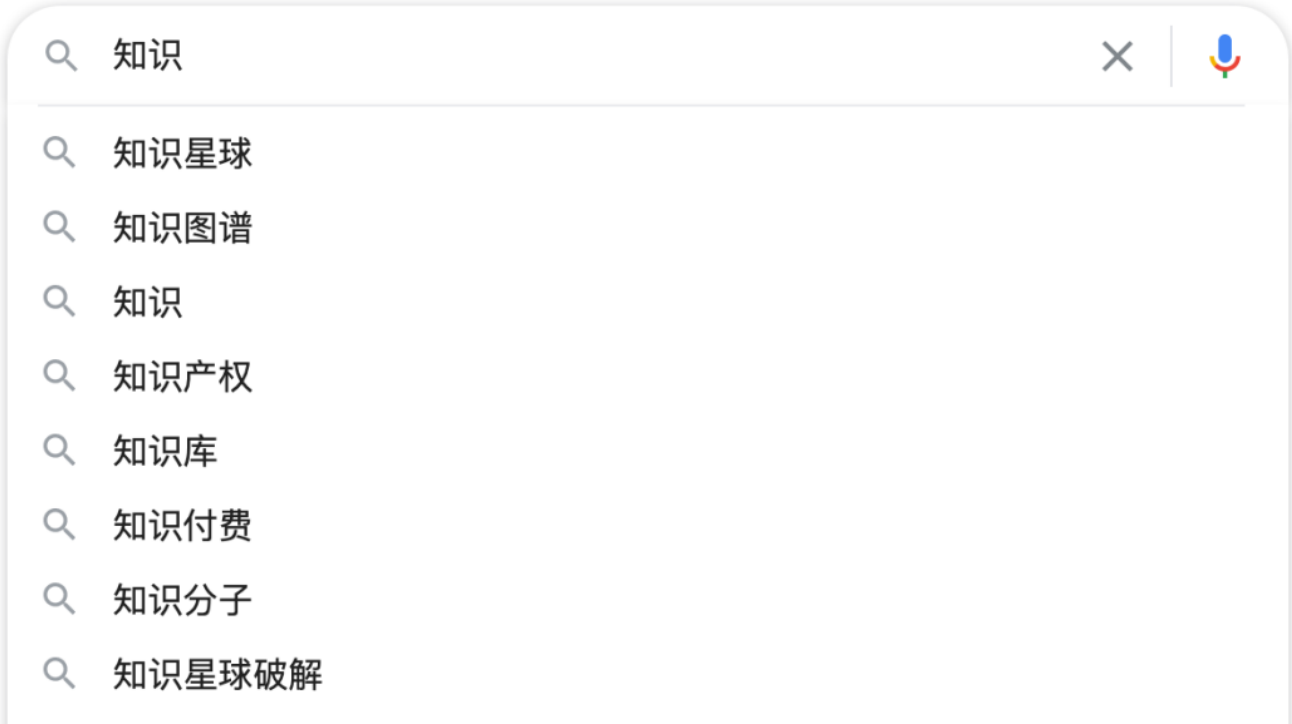
对原始Query首先需要做若干预处理，包括必要的**纠错、补全**，从Query中筛选出需要进行扩展的**主题词或实体词**，对最终的目标词完成扩展。这里我们着重关注 **Expansion Terms** 部分。

从哪里扩展

首先考虑扩展词从哪里来，这点上的思路和大部分语义相关的NLP任务类似，想法其实很直白，要不从**用户习惯、行为**中来，要不从描述**事物本身语义**中来：

一、业务场景语境

从用户的搜索log中可以挖掘出大量搜索词的固定搭配



这些词首先保证了与原始Query较高的匹配度（都包含“知识”这个词），在业务上也满足大多数用户关注的话题。缺点是这一类扩展词仅来自于统计学层面，与语义无关，无法满足语义层面上的相关性。

二、文档语料

除了来自用户的搜索Query，文档库也是扩展词的重要来源。丰富的语境能够提供词之间的相互关系

中国癌症地图
graphpad折线图
肺炎治疗
分层聚类热图
医生工资
入门科研
心衰指南
sci论文
科研论文

这类扩展词的缺点也比较明显，它与原始Query存在文本上的差异较大，增加召回的同时，可能会牺牲一定的匹配度。

三、构建领域知识库

最后一个途径是构建特定的领域知识库，优点是对语义相关度可以做更精准的控制，但是构建成本较高，同时如何将知识信息融入到原统计机器学习的算法中也有不小难度。

当然，理想的形态必然是混合以上三种来源，取其优点，可以观察谷歌的相关搜索：

知识蒸馏的相关搜索

知识蒸馏bert	Distilling the Knowledge in a Neural Network
知识蒸馏github	蒸馏 方法
知识蒸馏目标检测	Soft target loss
Keras 知识蒸馏	KD loss
在线蒸馏	Distill paper

融合了字面、语义、知识多方面的相关性。

扩展模型思路

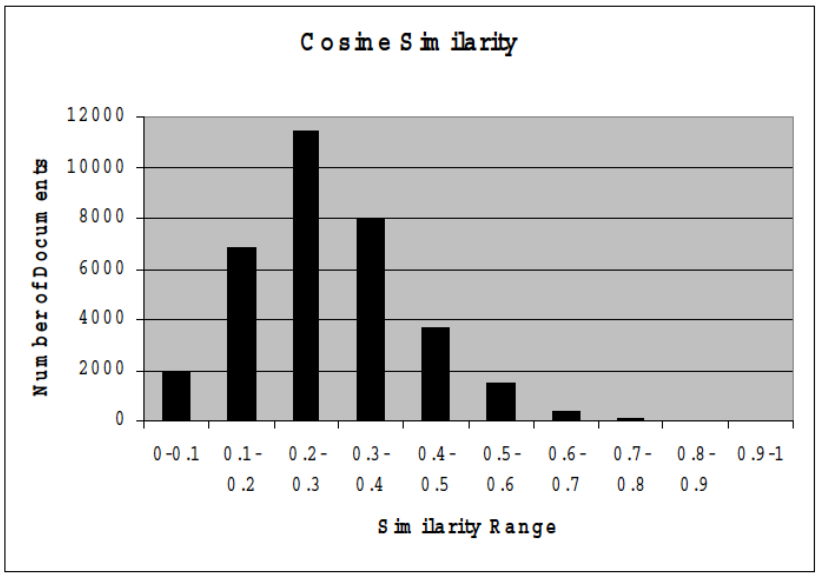
到模型部分，主要处理的就是原始Query中的term与待扩展的term/phrase如何产生关联。目前主流方案为两大类，一类是以贝叶斯模型为核心，统计扩展term与Query之间的条件概率。另一类的思路是把问题抽象成一个翻译模型，将Query中的词从scr到target语言完成改写。

一、相关模型

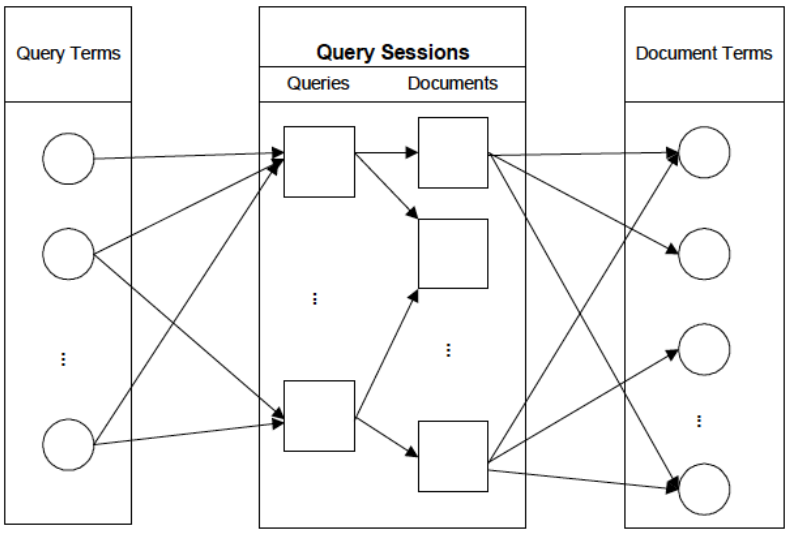
这种方案其实特别直白，Query词与扩展词在语料中**共现值**越大，说明它们相关性越高。也很容易想到使用**TF-IDF**一类的方式去做。经典的文献可以看这篇早在02年发表的工作：

《Probabilistic Query Expansion Using Query Logs》

作者提出需要做Query扩展的原因是认为用户输入的Query词与实际文档集中的词存在差异，所以在传统BM25算法搜索的过程中很有可能无法命中。下图是作者做的验证工作，将文档和Query都使用词袋向量表示，向量中元素值为TF-IDF，可以看到峰值区间对应的相似度并不高。



需要构建这种联系很自然的方式就是利用用户行为日志数据，用贝叶斯模型构建概率分布：



最终公式可以表示为：

$$P(w_j^{(d)} \mid w_i^{(q)}) = \sum_{\forall D_k \in S} (P(w_j^{(d)} \mid D_k) \times \frac{f_{ik}^{(q)}(w_i^{(q)}, D_k)}{f^{(q)}(w_i^{(q)})})$$

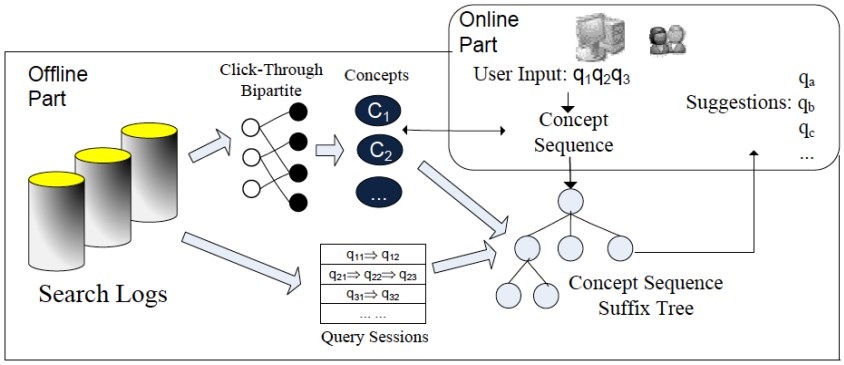
式子右侧括号中分别融合了扩展词在文档集中的**先验概率**、Query词与文档在交互session中的**共现频数**以及Query词在交互session中的**频数**。

虽然这篇文章过去了将近20年，但是后续的利用相关模型的算法都逃不出这个套路，方法简单却有效，在刚着手做该任务时不妨可以选它作为baseline。

往后大家对于这个套路的Query扩展优化，多关注于提高扩展词的质量。一个很自然的逻辑就是可以用term作为扩展词，同样地，phrase或concept短语也可以。

《Context-Aware Query Suggestion by Mining Click-Through and Session Data》

比如类似这篇文章中，因为Query中出现的term与它们对应的主题是多对多的，作者通过Query聚类挖掘出潜在的几种意图concept词，结合考虑Query Session的问题再进行后续扩展。

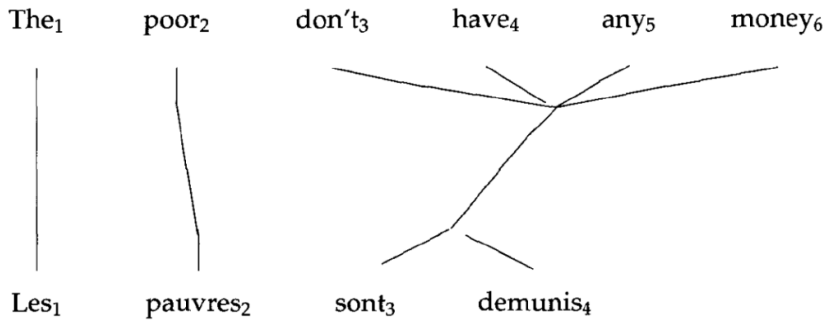


二、翻译模型

除了挖掘出扩展词的方案，另一个方向是对Query词进行直接改写。仍然从最简单的开始，各种复杂模型的起点其实都源自符合人类直觉的简单假设。

《The Mathematics of Statistical Machine Translation: Parameter Estimation》

这是篇年代更加久远的文章，发表于1993年，它就是著名的 IBM算法 。本身与Query扩展无关，主要工作是做机器翻译，但是它阐述了翻译模型最原始的假设：



完成翻译，我们要完成的无非是两件事：1）给定一个待翻译的句子，返回目标语言表达这个句子各个词意思的词；2）将原始语言的词与目标语言的词一一对应（alignment）。

而 IBM算法 的核心就是把这个问题抽象成“对齐”分布式是一个隐变量的概率问题：

$$\Pr(\mathbf{f},\mathbf{a}|\mathbf{e}) = \Pr(m|\mathbf{e})\prod_{j=1}^{m}\Pr(a_j|a_1^{j-1},f_1^{j-1},m,\mathbf{e})\Pr(f_j|a_1^{j},f_1^{j-1},m,\mathbf{e}).$$

利用 EM算法 完成求解即可。

《Towards concept-based translation models using search logs for query expansion》

做Query扩展时，这一思想也被迁移过来：

$$P(w|Q) = \sum_{q\in Q} P(w|q)P(q|Q)$$

$$P(D|Q,\theta) = \frac{\varepsilon}{(I+1)^J}\prod_{w\in\mathbf{d}}\sum_{q\in\mathbf{q}} P(w|q,\theta).$$

连公式的形式都与原 IBM算法 是一致的，式子中theta就是改写操作中原始词与目标词的对齐概率参数。文章中，作者也进一步实验了phrase和concept词的结果。同时，配合 term weighting 一起食用，效果更好喔！

《Learning to Rewrite Queries》

再后续，大家会考虑进一步优化alignment分布的学习以及融入更多的语义特征进去，毕竟 IBM算法 仅从统计词频的角度估算分布还是太过单薄。近年来深度学习的发展自然就带动一些传统模型向神经网络向的方法上迁移：

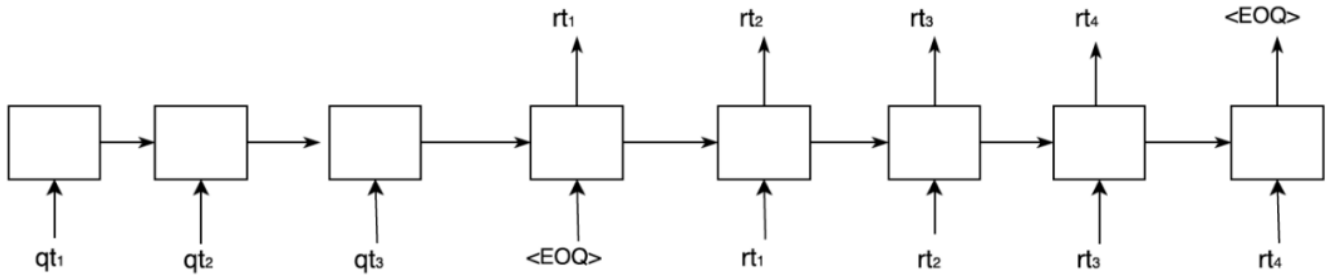
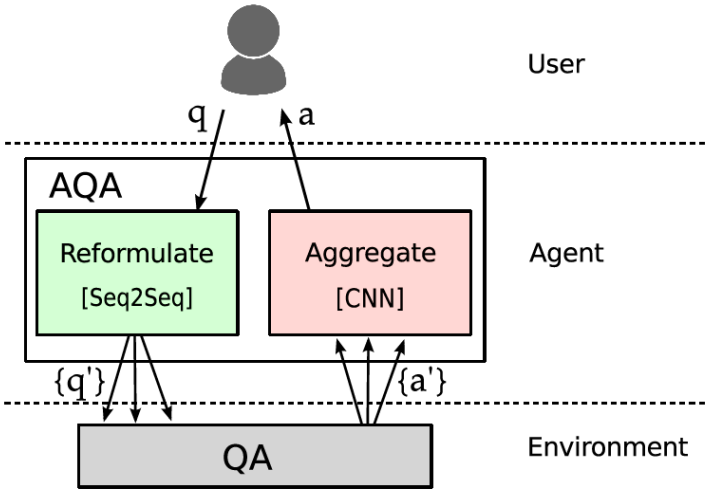


Figure 2: Scheme of Sequence to Sequence LSTM for Generating Query Rewrite

《Ask the Right Questions: Active Question Reformulation with Reinforcement Learning》

谷歌在 2018 ICLR 上发表的工作合并了使用序列模型完成Query改写，考虑使用强化学习来进一步增强



这个方案的大致思路是，模型与索引系统连接，若改写后的Query可以索引出排序更靠前的内容，则给予强化模型正向的激励。而且，train好的强化模型也可以倒过来finetune改写模型。

增强语义相关性

可以看到，计算原始Query到扩展词或改写词的关联已经有非常多成熟的方法，甚至可以在自己场景里设计比较tricky的强化模型方案。而我们认为进一步提高效果的关键，还是需要主动对**业务内容的组织、理解、良好的结构化**。目前火热的各种文本预训练模型、知识图谱等都印证了这一点。常常看到有人看衰知识图谱发展，不可否认因为它没有一个大而独立的场景，所以它必然无法像CV那样大放光芒。但是我们一直认为知识图谱的技术目前最合理的使用方案是浸润在日常的各个技术中，它是帮助老技术突破瓶颈的途径，没有新东西，听起来当然不够性感。

说到这里，我们对语义相关性的增强就可以利用在往期标签生成的文章中提到的方法

《A User-Centered Concept Mining System for Query and Document Understanding at Tencent》

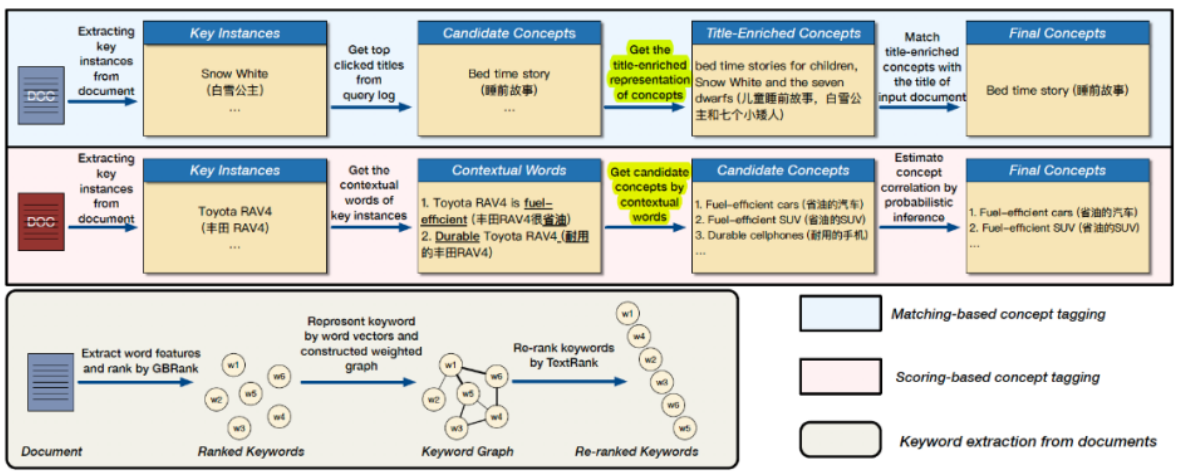
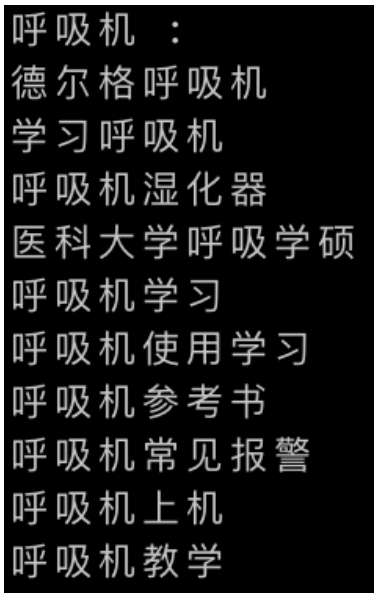


Figure 3: The overall procedures of concept tagging for documents. We combine both a matching-based approach with a scoring-based approach to handle different situations.

利用大量Query的点击数据，挖掘出该场景下的各种concept词。进一步，我们将concept词与医学知识图谱形成关联，从而替代了前面提到的Query聚类方案。

简单实践

挖掘出高质量的Concept词后，由上文提到的关联模型就可以获得不错的效果：



当遇到多个实体，使用关联打分即可：

$$CoWeight_Q(w_j^{(d)}) = \ln\Big(\prod_{w_t^{(q)} \in Q} (P(w_j^{(d)} \mid w_t^{(q)}) + 1)\Big)$$

总结

总的来说，Query扩展本身并不算一个复杂的工作，想要最终效果做得好，我们的建议是：**把复杂的工作向后撤。**生成模型或者强化模型看起来很fancy，操作门槛以及工程上的支持难度都较大。踏踏实实做好数据结构化（知识图谱构建、知识表示学习、长文本标签化等等），在下游应用里，用简单模型就能看到效果。有了baseline之后，我们会考虑用复杂的方案把关联性构建得更好。

参考文献

[1]. Probabilistic query expansion using query logs
[2]. Ask the Right Questions- Active Question Reformulation with Reinforcement Learning
[3]. Concept-Based Interactive Query Expansion