全面理解搜索Query：当你在搜索引擎中敲下回车后，发生了什么?

DSM 艾达AI 2020-11-03

原文来源于“腾讯技术工程”知乎官方账号，由艾达AI解读整理。希望大家读完这篇文章后能对query有一个基本全面的认识。

本文主要介绍在搜索中的query理解，会相对系统性地介绍query理解中各个重要模块以及它们之间如何work起来共同为搜索召回及排序模块服务。

原文地址：https://zhuanlan.zhihu.com/p/112719984

引言

Query理解（QU，Query Understanding），简单来说就是从词法、句法、语义三个层面对query进行结构化解析。这里的query从广义上来说涉及的任务比较多，比如常见的基于结构化数据的KBQA问答系统、基于文本的机器阅读理解问答系统、基于问答对的FAQ问答系统、以及人机对话系统等。本文主要介绍在搜索中的query理解，会相对系统性地介绍query理解中各个重要模块以及它们之间如何work起来共同为搜索召回及排序模块服务。

搜索系统相关知识

一个基本的搜索系统大体可以分为离线挖掘和在线检索两部分，其中包含的重要模块主要有：item内容理解、query理解、检索召回、排序模块等。

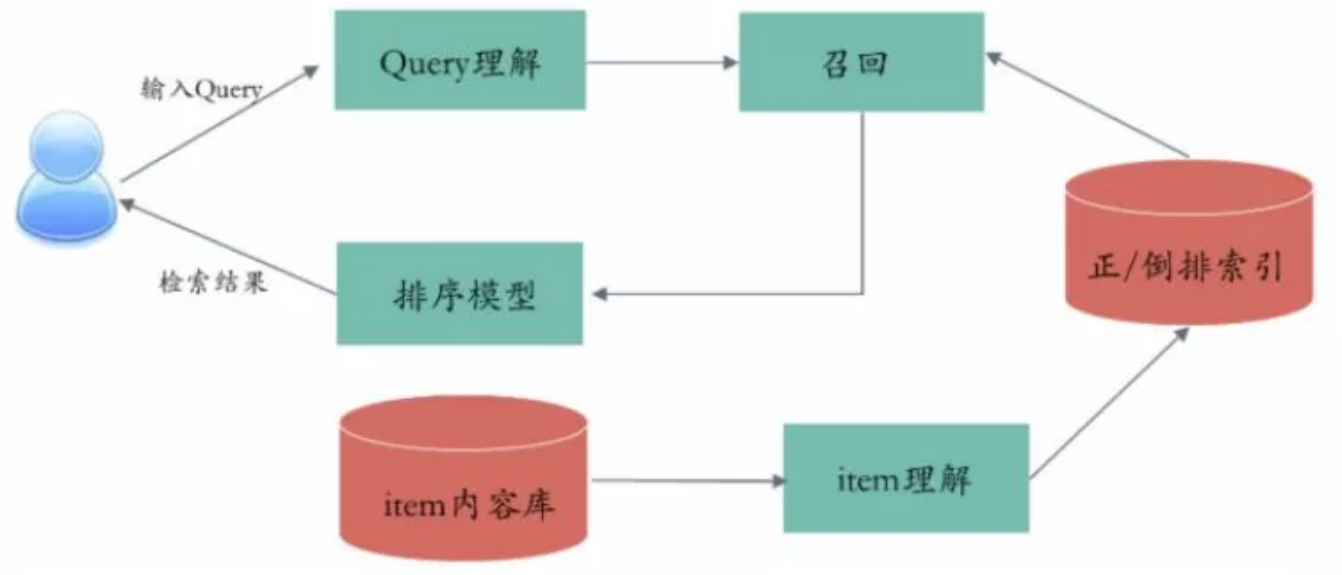

图1.搜索系统基本架构

离线挖掘

离线侧的基础工作包括：item内容的获取、清洗解析、item内容理解、构建排序模型样本及特征工程等。进行item内容理解之后，对相应的结构化内容执行建库操作，分别构建正排和倒排索引库。

除了基本的文本匹配召回，还需要通过构建query意图tag召回或进行语义匹配召回等多路召回来提升搜索语义相关性以及保证召回的多样性。

在线检索

线上执行检索时大体可以分为基础检索（BS）和高级检索（AS）两个过程，其中BS更注重term级别的文本相关性匹配及粗排，AS则更注重从整体上把控语义相关性及进行精排等处理。以下面这个框图为例，对于从client发起的线上搜索请求，会由接入层传入SearchObj（主要负责一些业务逻辑的处理），再由SearchObj传给SearchAS（负责协调调用qu、召回和排序等模块），SearchAS首先会去请求SearchQU服务（负责搜索query理解）获取对query理解后的结构化数据，然后将这些结构化数据传给基础召回模块BS，BS根据切词粒度由粗到细对底层索引库进行一次或多次检索，执行多个索引队列的求交求并拉链等操作返回结果。

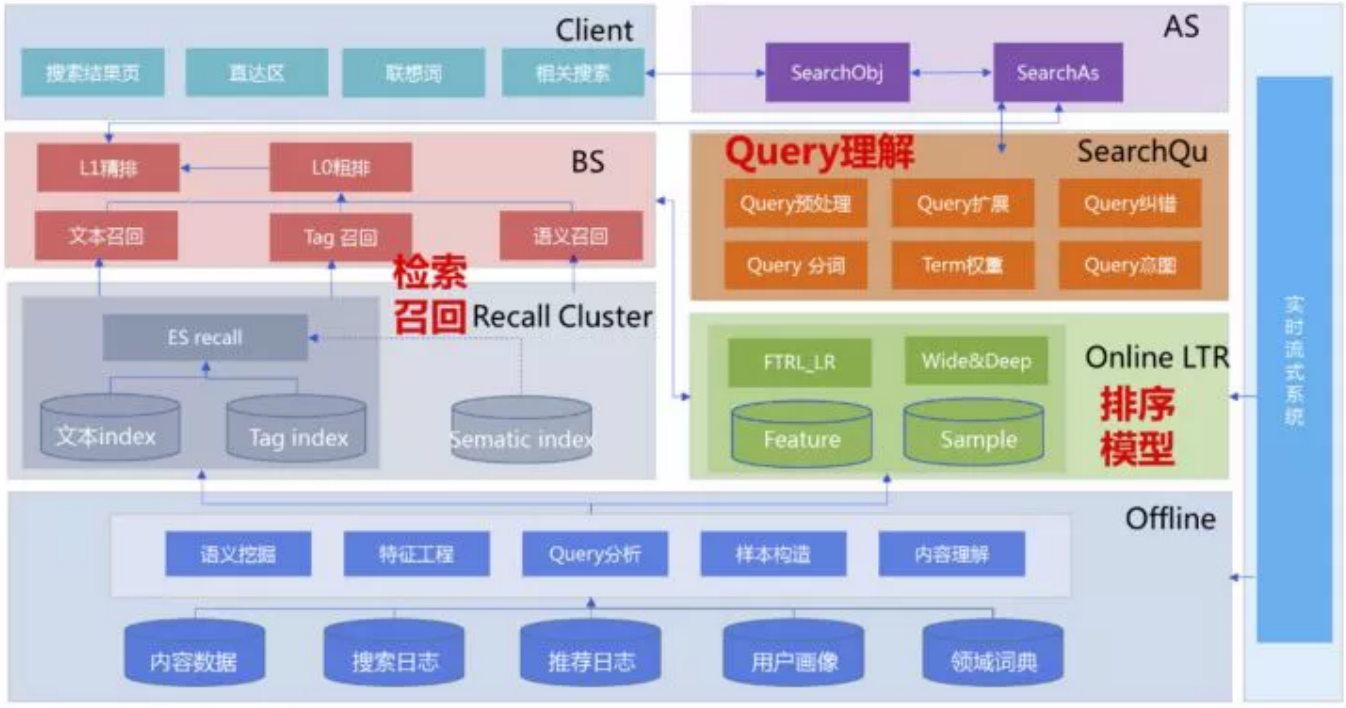

图2.在线检索流程

同时BS还需要对文本、意图tag、语义召回等不同路召回队列根据各路召回特点采用多个相关性度量（如：BM25文本相似度、tag相似度、语义相关度等）进行L0粗排截断以保证相关性，然后再将截断后的多路召回进行更精细的L1相关性融合排序，更复杂一些的搜索可能会有L0到LN多层排序，每层排序的侧重点有所不同，越高层次item数变得越少，相应的排序方法也会更复杂。

BS将经过相关性排序的结果返回给SearchAS，SearchAS接着调用SearchRank服务进行ctr/cvr预估以对BS返回的结果列表进行L2重排序，并从正排索引及摘要库等获取相应item详情信息进一步返回给SearchObj服务，与此同时SearchObj服务可以异步去请求广告服务拉取这个query对应的广告召回队列及竞价相关信息，然后就可以对广告或非广告item内容进行以效果（pctr、pcvr、pcpm）为导向的排序从而确定各个item内容的最终展示位置。

Query理解

目前业界的搜索QU处理流程还是以pipeline的方式为主，即拆分成多个模块分别负责相应的功能实现，pipeline的处理流程可控性比较强，但存在缺点就是其中一个模块不work或准确率不够会对全局理解有较大影响。为此，直接进行query-item端到端地理解如深度语义匹配模型等也是一个值得尝试的方向。

Pipeline流程

搜索query理解包含的模块主要有：query预处理、query纠错、query扩展、query归一、联想词、query分词、意图识别、term重要性分析、敏感query识别、时效性识别等。以下图为例，线上来一个请求query，为缓解后端压力首先会判断该query是否命中cache，若命中则直接返回对该query对应的结构化数据。若不命中cache，则首先会对query进行一些简单的预处理，接着进行query分词并移除一些噪音符号，然后进行query纠错处理。

对query纠错完后可以做query归一、扩展及联想词，用于进行扩召回及帮助用户做搜索引导。接着再对query分词后的term做重要性分析及紧密度分析，对无关紧要的词可以做丢词等处理，有了分词term及对应的权重、紧密度信息后可以用于进行精准和模糊意图的识别。除了这些基本模块，有些搜索场景还需要有对query进行敏感识别、时效性分析等其他处理模块。

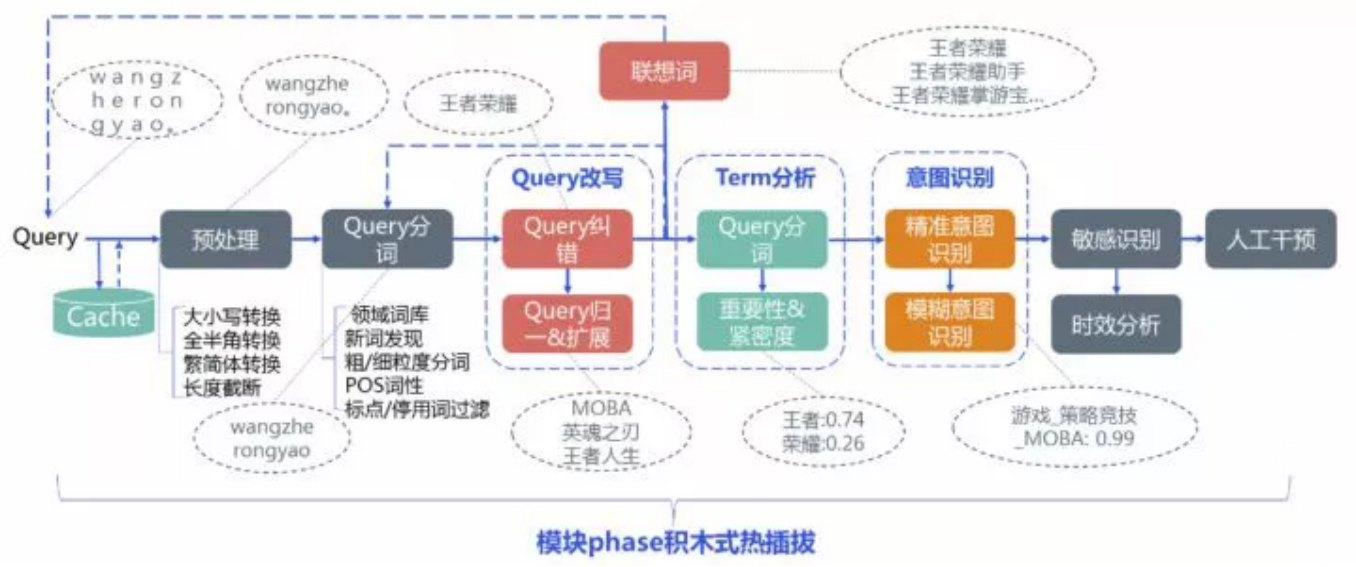

图3.Query理解包含的主要模块

Query分词

在搜索中query切词一般会做粒度控制，分为细粒度和phrase粗粒度两个级别，在进行召回时可以优先用phrase粗粒度的切词结果进行召回能得到更精准相关的结果，当phrase粗粒度分词进行召回结果不够时，可以采用拆分后的细粒度分词进行二次重查扩召回。

紧密度分析

Term紧密度，主要用于衡量query中任意两个term之间的紧密程度，如果两个term间紧密度比较高，则这两个term在召回item中出现的距离越近相对来说越相关。

以搜索query“下载深海大作战”为例，经分词工具可能切分成“下载深海大作战”，但其实“大”和“作战”的紧密度很高，从文本相关性角度来看，召回“喵星大作战”app要一定程度比“大人物作战”会

更相关。当然，在query中并不是两个term的距离越近紧密度越高，可以通过统计搜索log里term之间的共现概率来衡量他们的紧密程度。

在进行召回时，传统的相关性打分公式如BM25TP、newTP、BM25TOP等在BM25基础上进一步考虑了term proximity计算，但主要采用两个term在item中的距离度量。有了query中的term紧密度，在召回构造查询索引的逻辑表达式中可以要求紧密度高的两个term需共同出现以及在proximity计算公式中融合考虑进去，从而保证query中紧密度高的两个term在item中出现距离更近更相关。

Term重要性分析

考虑到不同term在同一文本中会有不一样的重要性，在做query理解及item内容理解时均需要挖掘切词后各个term的重要性，在进行召回计算相关性时需要同时考虑query及item侧的term重要性。对于重要级别最低的term可以考虑直接丢词，或者在索引库进行召回构造查询逻辑表达式时将对应的term用“or”逻辑放宽出现限制，至于计算出的term重要性分值则可以用于召回时计算相关性。

其中item内容侧的term重要性可以采用LDA主题模型、TextRank等方法来挖掘，至于query侧的term重要性，可以把它视为分类或回归问题，通过训练svm、gbdt等传统机器学习模型和精细的特征工程可以得到较好的结果。还有的方法就是利用深度学习模型来学习term重要性，比如通过训练基于BiLSTM+Attention的query意图分类模型或基于eq2Seq/Transformer训练的query翻译改写模型得到的attention权重副产物再结合其他策略或作为上述分类回归模型的特征也可以用于衡量term的重要性。

搜索引导

除了保证搜索结果的相关性，一个完善的搜索引擎还需要给用户提供一系列搜索引导功能，以减少用户的搜索输入成本，缩短用户找到诉求的路径。搜索引导按功能的不同主要可以分为：搜索热词、搜索历史、query改写、搜索联想词。

Query改写

按照改写功能的不同，query改写可以分为query纠错、query归一、query扩展三个方向。其中query纠错负责对存在错误的query进行识别纠错，query归一负责将偏门的query归一变换到更

标准且同义的query表达，而query扩展则负责扩展出和源query内容或行为语义相关的query列表推荐给用户进行潜在需求挖掘发现。

Query纠错

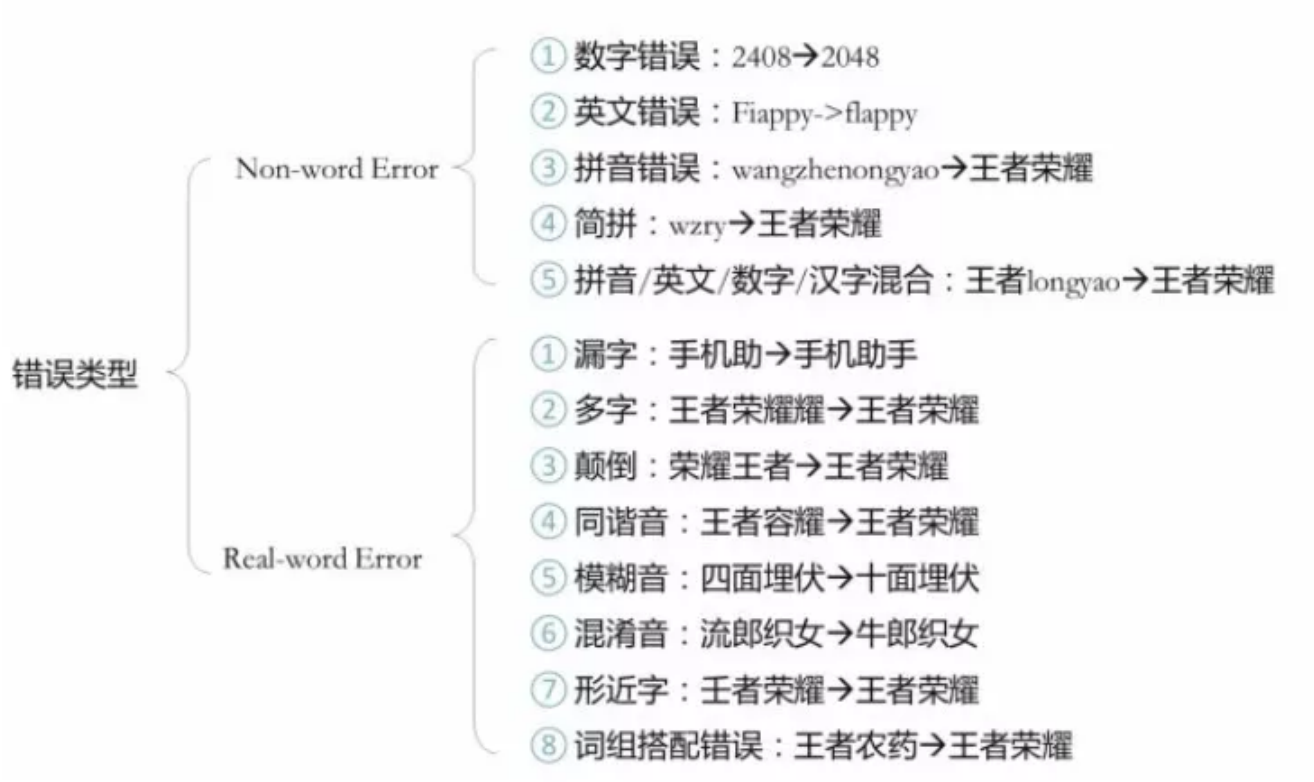

图4.Query纠错的主要错误类型

纠错任务主要包含错误检测和错误纠正两个子任务，其中错误检测用于识别错误词语的位置，简单地可以通过对输入query进行切分后检查各个词语是否在维护的自定义词表或挖掘积累的常见纠错pair中，进一步的做法是通过训练序列标注模型的方法来识别错误的开始和结束位置。

至于错误纠正，即在检测出query存在错误的基础上对错误部分进行纠正的过程，其主要包括纠错候选召回、候选排序选择两个步骤。

对于英文单词错误、多漏字、前后颠倒等错误可以通过编辑距离度量进行召回，编辑距离表示从一个字符串变换到另一个字符串需要进行插入、删除、替换操作的次数。

对于等长的拼音字型错误类型还可以用HMM模型进行召回，另外结合BERT等预训练语言模型来进行纠错候选召回是值得尝试的方向，如在BERT等预训练模型基础上采用场景相关的无监督语料继续预训练，然后在错误检测的基础上对错误的字词进行mask并预测该位置的正确字词。

Query扩展

Query扩展，即通过挖掘query间的语义关系扩展出和原query相关的query列表。Query列表的结果可以用于扩召回以及进行query推荐帮用户挖掘潜在需求。

我们可以通过挖掘搜索session序列和点击下载行为中query间的语义关系来做query扩展。

对于将query进行embedding向量化的方法，可以先离线计算好已有存量query的embedding表示，然后用faiss等工具构建向量索引，当线上有新的query时通过模型inference得到对应的embedding表示即可进行高效的近邻向量检索以召回语义相似的query。

Query归一

query归一主要对同近义表达的query进行语义归一的作用。一些用户的query组织相对来说比较冷门，和item侧资源的语义相同但文字表达相差较大，直接用于召回的话相关性可能会打折扣，这时如果能将这些query归一到相对热门同义或存在对应资源的query会更容易召回相关结果。

离线阶段，从搜索点击中先挖掘出语义表达相近的query-query、item-item或query-item短语对，然后再将语义相近的query/item短语对进行语义对齐，语义对齐后从中抽取出处于相同或相近上下文中的两个词语作为同义词对候选。

线上对query进行归一时，则和离线同义词挖掘的过程相反，对query进行分词后读取线上存储的同义词表做同义词候选替换，对替换网络进行对齐生成候选query，最后通过结合语言模型概率及在当前上下文的替换概率或者构造特征训练GBDT等模型的方式对候选query进行排序得到最终的归一query。

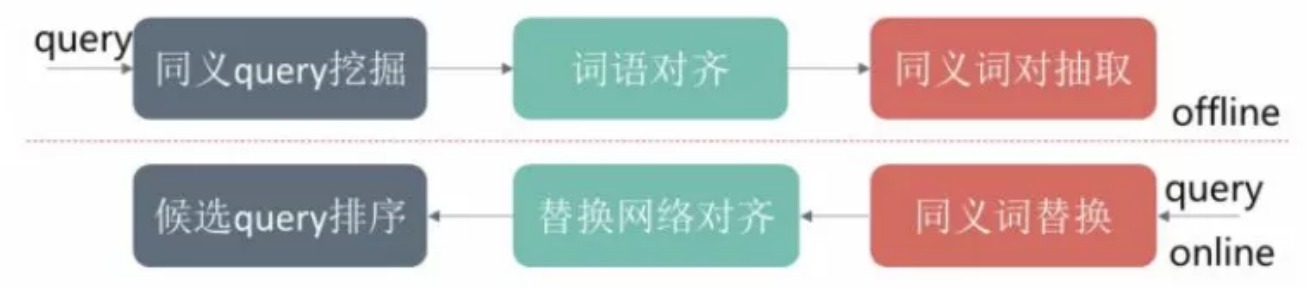

图5.Query同近词归一