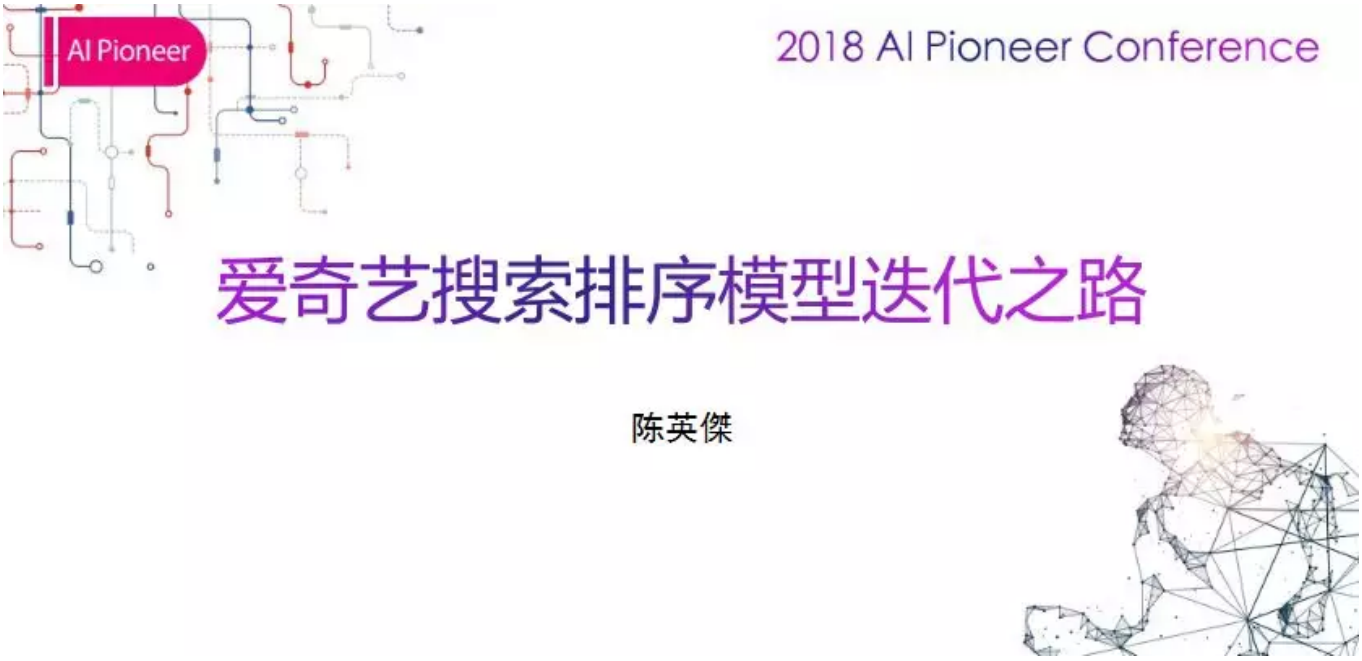
「回顾」爱奇艺搜索排序模型迭代之路

原创 陈英傑 DataFunTalk 2018-12-21



> 分享嘉宾：**陈英傑 爱奇艺 研究员**

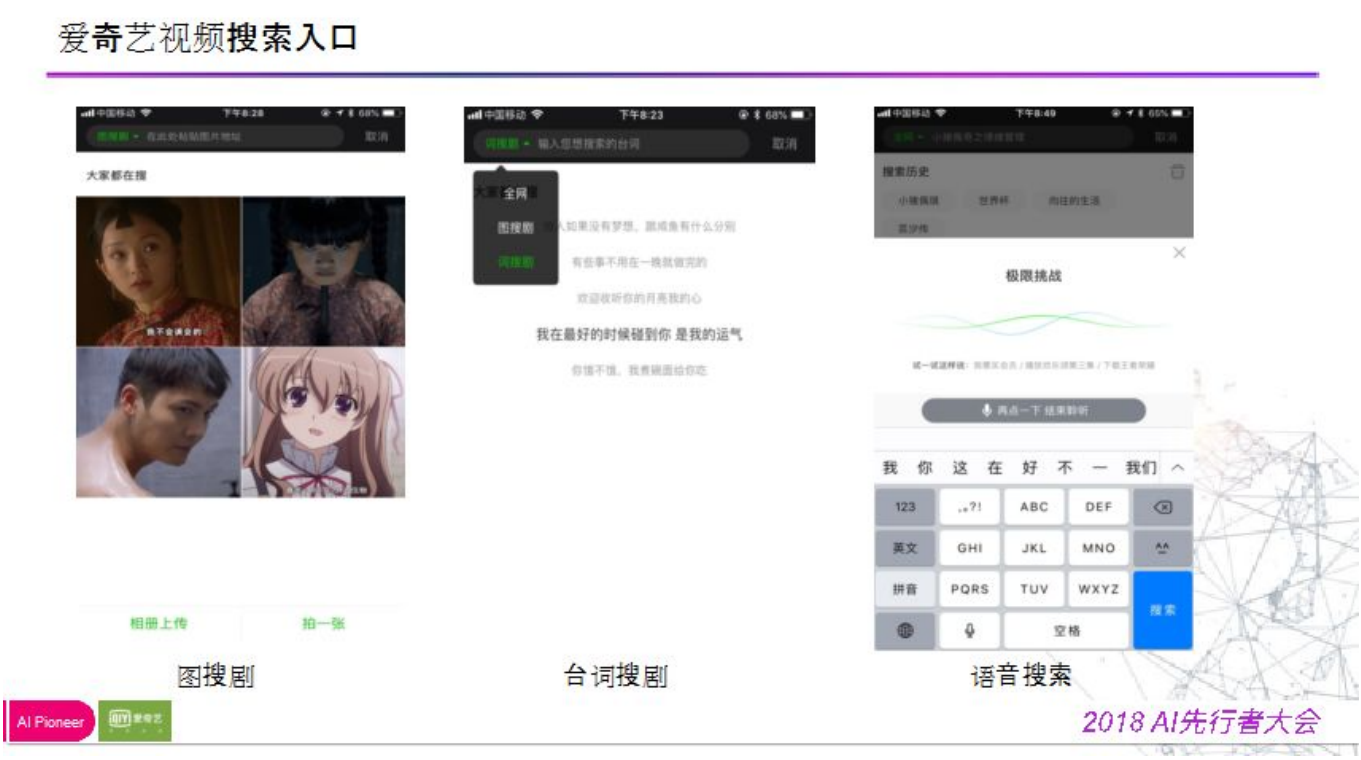
> 编辑整理：**孙锴**
> 内容来源：**AI先行者大会《爱奇艺搜索排序模型迭代之路》**
> 出品社区：**DataFun**
> 注：欢迎转载，转载请注明出处。

一、摘要

本次分享内容为爱奇艺在做视频搜索时，遇到的真实案例和具体问题；以及面对这些问题的时候，我们的解决方案。这次分享的ppt针对一线的开发人员，希望可以给一线的开发人员提供一些启示。

二、介绍

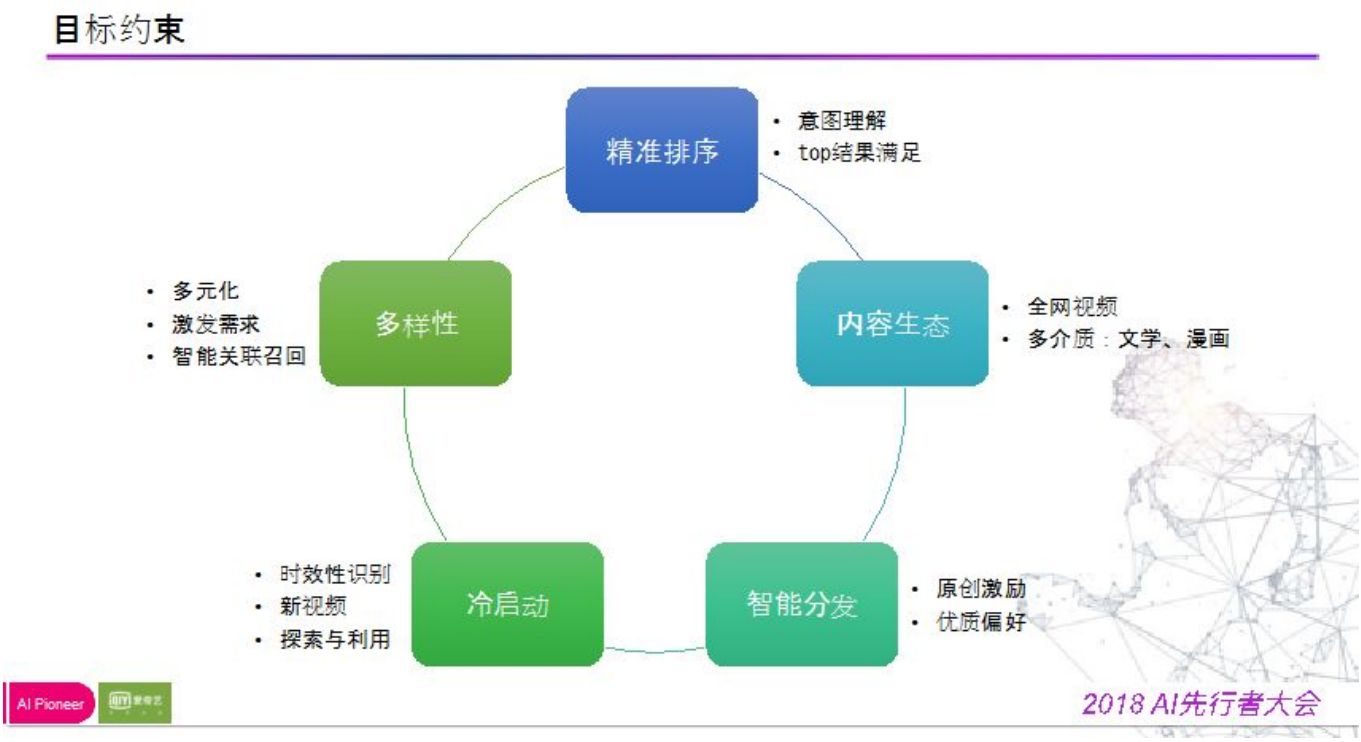
首先介绍一下我们支持的搜索入口，在我们app的搜索框里，支持下图所示的搜索方式：图搜索、台词搜索、语音搜索。



今天分享的，其实还是一个更加通用的搜索方式，即文本查询。通过把用户输入的文本做自然语言处理后进行的关键字查询，在此，我们做了很多自然语言处理和语义的理解。

在视频内容层面，最重要的是视频本身的描述信息，如标题，演职人员等信息。还有一个是内容的语义，我们当前并不是多模态特征去抽取，更多的是通过用户对该视频的观看行为，如搜索、浏览、评论、弹幕等各种行为，由此产生的数学结构去抽取语义信息。所以我们今天更关注在doc层和query层是如何做这些匹配的。

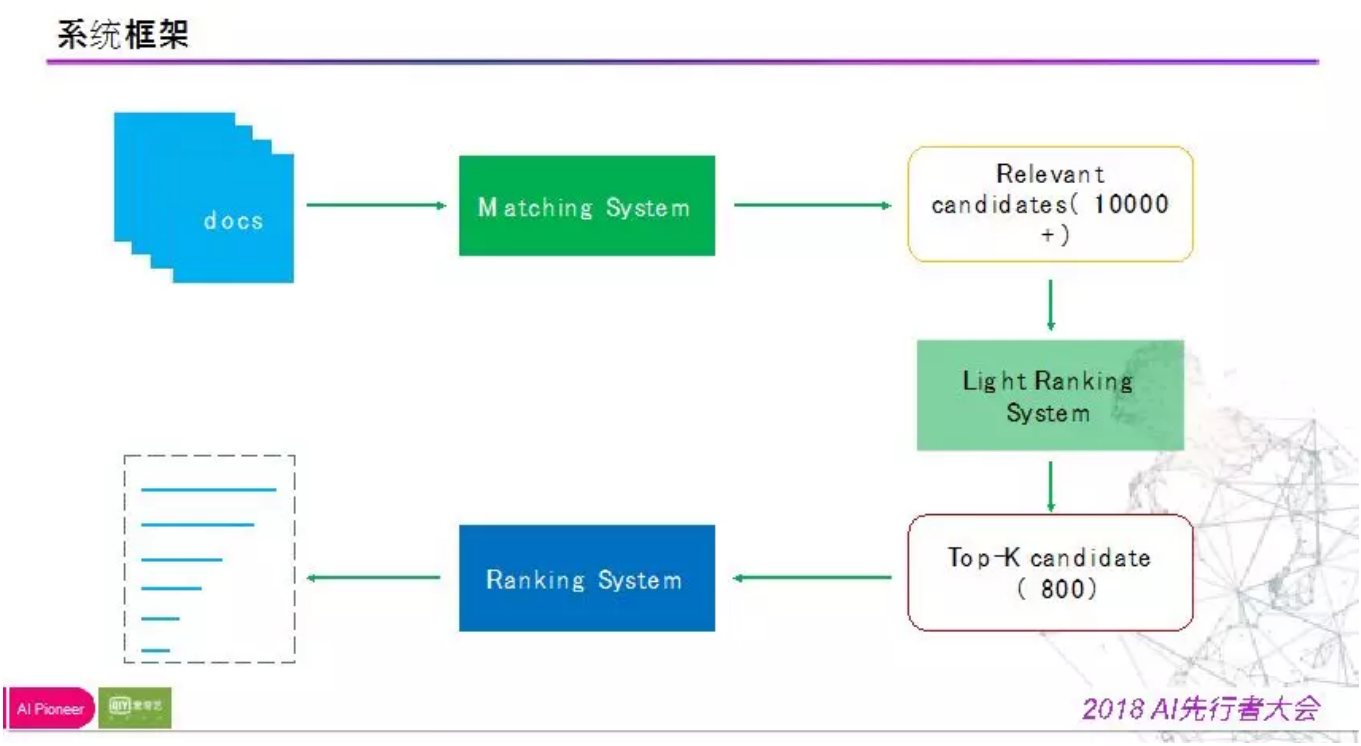
在介绍具体的匹配过程之前，我们先了解一下一个通用的搜索系统的约束条件（下图所示）：



> **1) 精准匹配**，用户搜什么词，展现什么内容，而且需要top结果排序；
> **2) 内容生态**，爱奇艺的视频搜索不仅仅是站内搜索引擎，而是全网的视频搜索引擎，所以我们会囊括所有中文的视频资源，包括我们没有版权的视频，我们希望打造的是，帮助用户链接到想要的资源，同时我们还支持文学、漫画资源的搜索；
> **3) 智能分发**，搜索结果有不同的版权方，我们需要对原创结果进行激励，防止略币驱逐良币，载流量上给优质资源进行扶持；
> **4) 冷启动问题**，新视频相比于老视频在特征上相对弱势，我们需要给予冷启动空间，在此做一些探索和利用；
> **5) 搜索多样性**，防止靠前结果都是一样的。

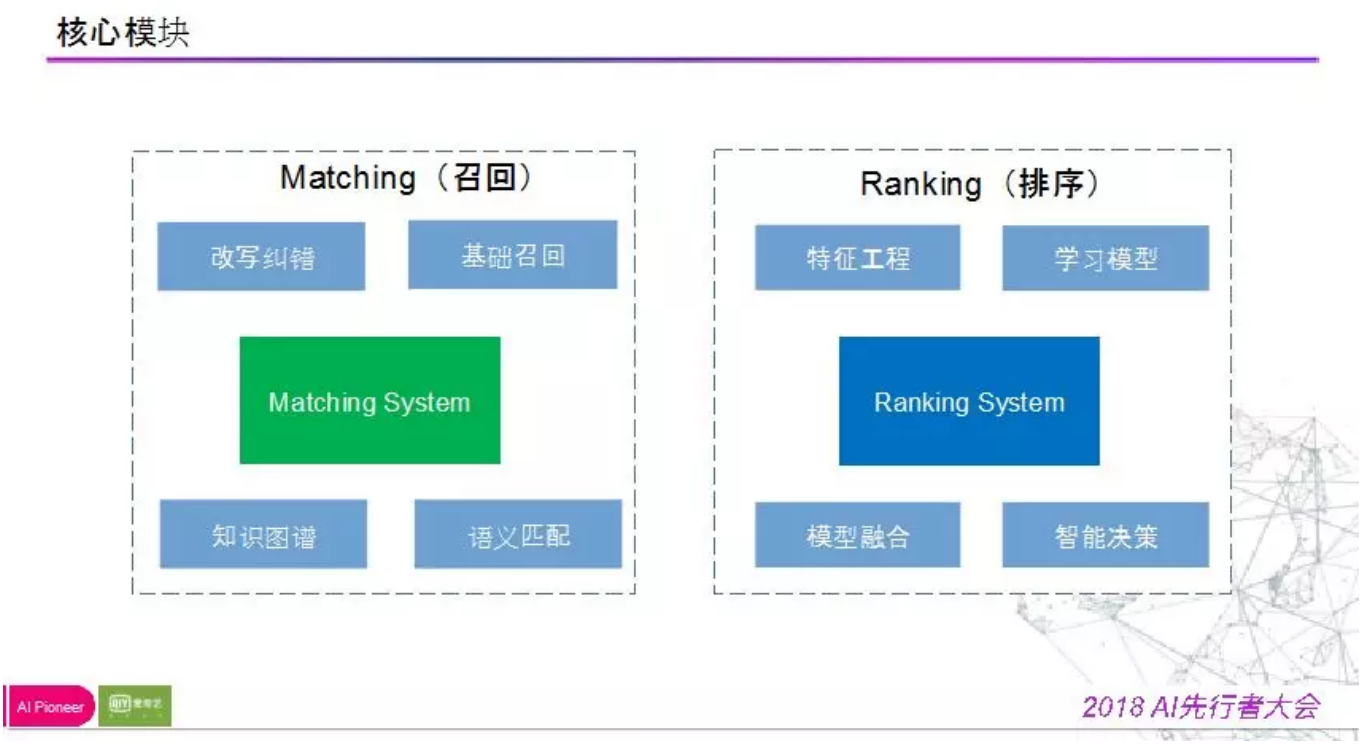
此外，我们发现，当用户在搜索产生的结果使得自身的的主需求得到满足的时候，可以激发用户一些其他的语义相关的结果。

在这样五个约束条件下，我们如何搭建全网的搜索引擎呢？下图即是我们的整体系统框架。



我们有大量视频资源，通过召回系统，即基于文本匹配的matching system，得到候选集，经过粗排和精排，最后返回给用户，这是大体的流程图，其中最重要的是召回系统和排序系统。

两个系统的重要模块很多，下图列举其中一些：包括改写纠错，基础召回，知识图谱召回，语义匹配召回；排序模块关注特征工程，学习模型的选择，模型融合与智能决策。



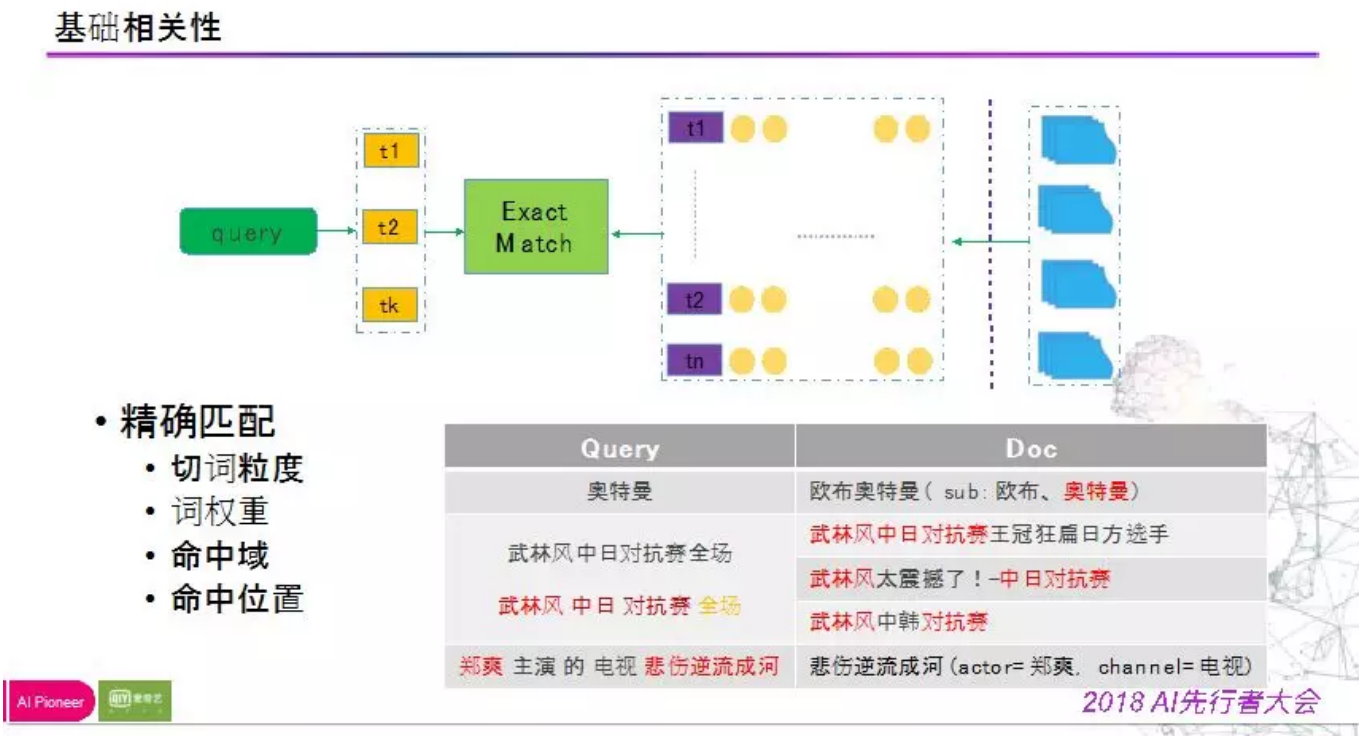
在此我们将要展开的是我们是如何进行一步步迭代的。

> **第一**， 召回策略的迭代，我们从基础相关性慢慢走到语义相关性的路径；
> **第二**， 排序模型的尝试。

三、 召回策略迭代

1. 基础相关性

首先是基础相关性，搜索引擎处理流程图如下：



通过对用户的query进行切词，将右边的视频资源的文本描述信息构建构建倒排索引，此过程为精确匹配过程，词匹配则倒排索引拉回归并，然后返回用户，此过程较为经典，在传统的搜索引擎也是比较成熟的应用方式。

这样一个流程里面，它解决的问题也比较通用：

> **1) 切词粒度**，不同的词的粒度会影响你是否可以通过倒排索引召回内容；
> **2) 词权重**，一个query中，哪些词是重要的，哪些是不重要的，会影响你在相关性计算的时候的最终得分。

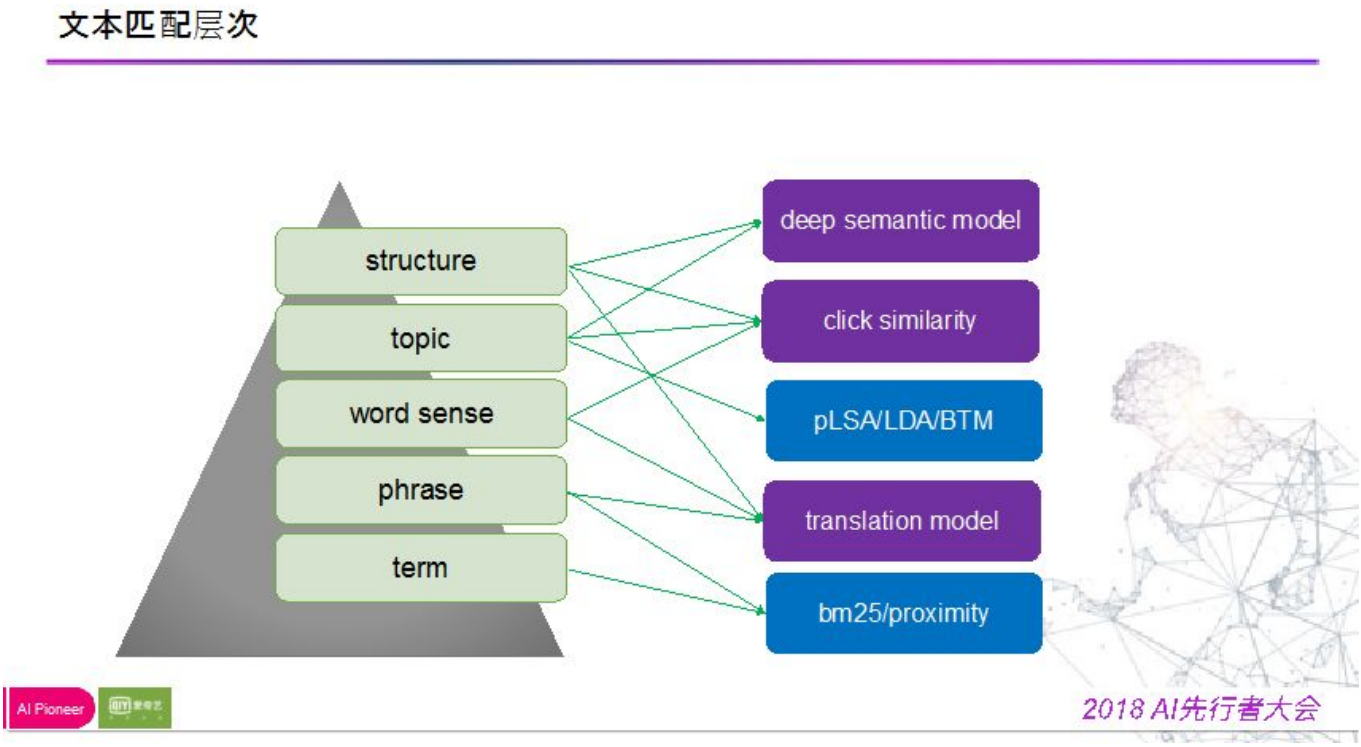
这其实是基础相关性中需要解决的问题，也是我们在1.0版本中，花了很多力气去解决的问题。需要注意的是，上面的问题并没有最优解，这是一个根据bad case不断做优化的过程。

这里举个例子，如下图（例子描述见视频）：

基础相关性解决不了的问题，我们归为四类：词汇的同义多以问题、语言表达差异、输入错误兼容、泛语义召回。

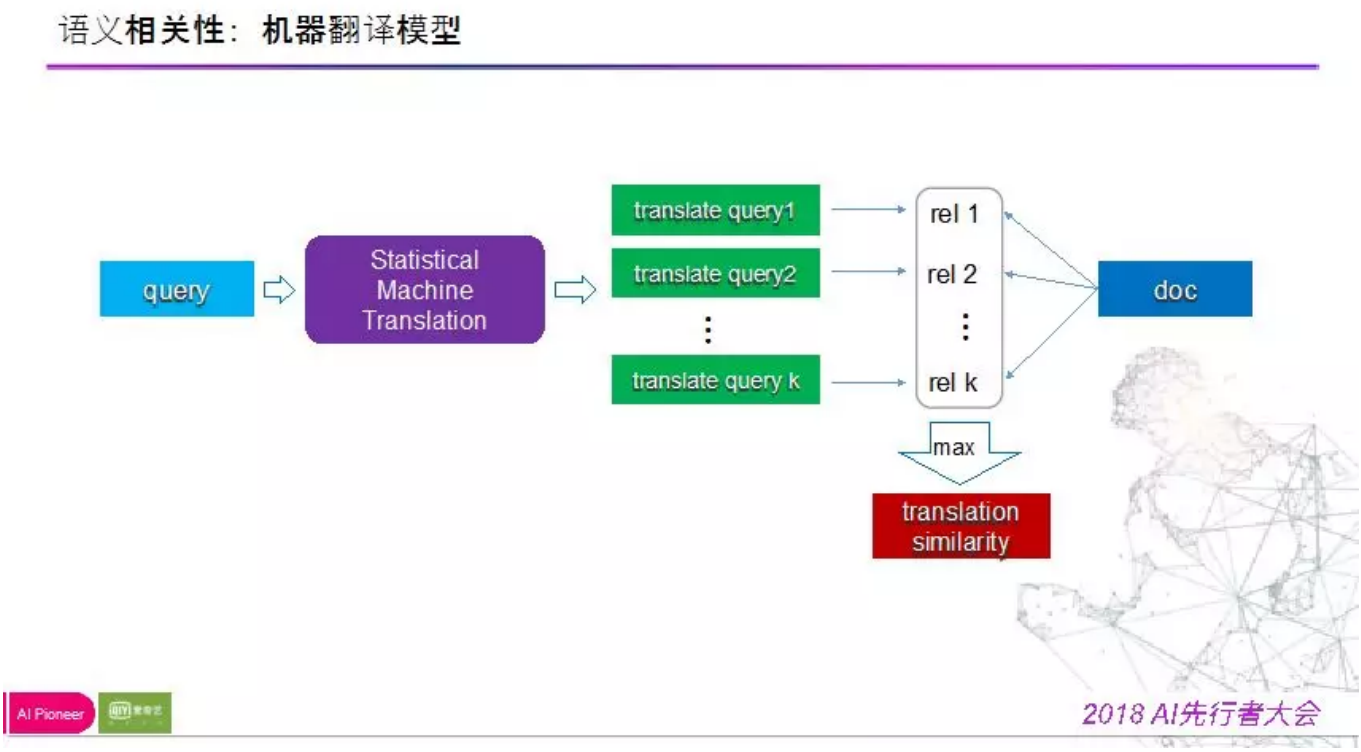
2. 语义相关性

在解决基础相关性遇到问题的时候，我们再来思考一下，在文本匹配上是怎么解决语义的问题，如下图所示：



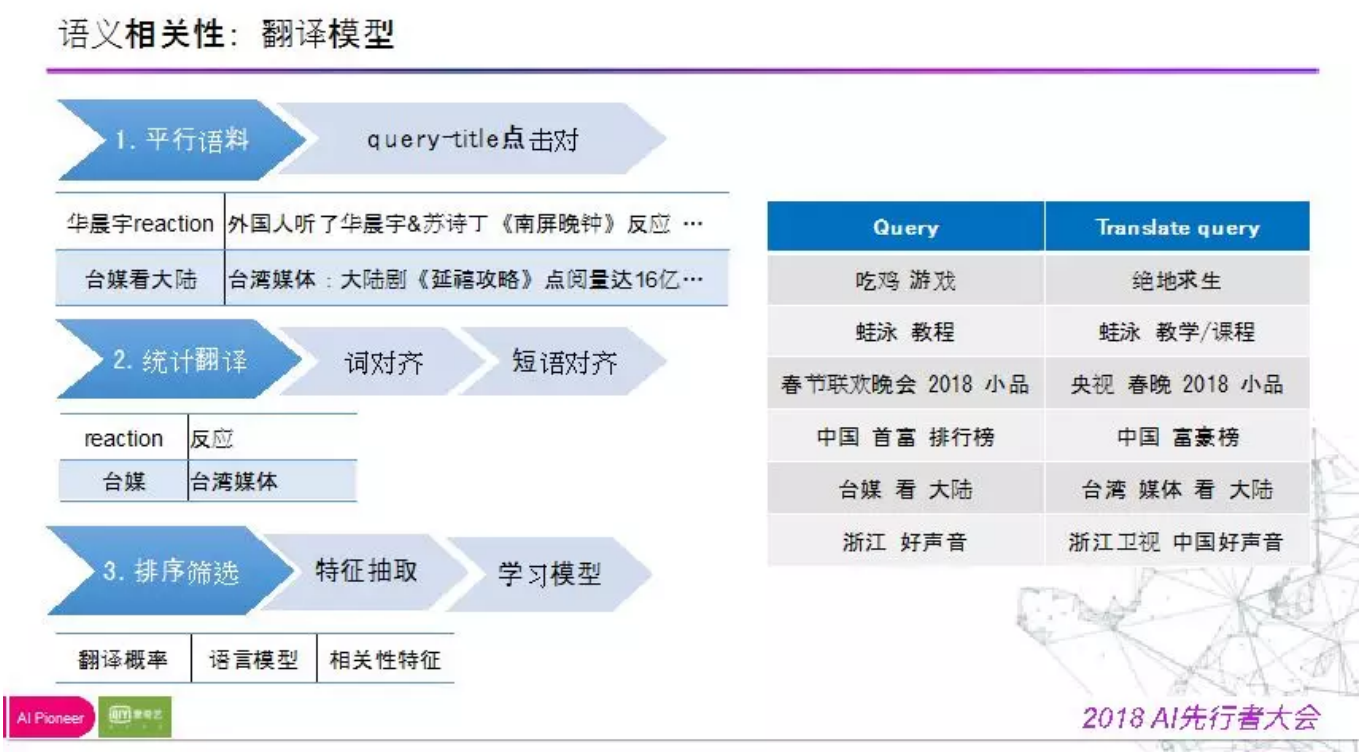
左边是XX老师整理的五个层面：词、词组、词义、主题、结构。在搜索场景下：我们有天然的用户搜索之后的点击行为，基于点击行为，我们可以在不同层面做语义匹配，紫色框是我们要解决的问题所用到的技术。

下图是机器翻译模型：



机器翻译是一个目前比较火热的领域，并且在深度学习出现之后，其准确率得到了飞跃性的提高。这是我们解决语义相关性的第一个手段。

2.1 翻译模型



由于用户在搜词的时候，并不会去把相关词汇都搜索一遍，这就需要由搜索引擎去拓展用户的查询词汇。通过翻译系统，可以将查询词转化成与语义需求相同的其他词汇，用这些扩展后的新词与视频做相关性计算，取top结果返回给用户，以此来实现拓展词召回。

具体策略为：

> **第一步**：根据用户的query以及点击的document生成doc-query点击对，以此来构建翻译的平行语料对；
> **第二步**：做词对齐和短语对齐，此时，我们并没有用到很多深度学习的技术，因为在搜索场景下，并不需要翻译结果的准确性，更为重要的是拓展出来的词汇是不是有意义的，是不是让这个系统往正向发展的；
> **第三步**：query中的词汇与拓展的词汇行程映射对，在映射对里会存在噪音，针对噪音，传统的基于统计的短语和基于词的翻译模型会存在一些问题，我们再从新标注一部分翻译的 ground truth。

在这个基础上，我们根据翻译模型给出的翻译概率，并通过语言模型判断翻译结果是否通顺，再结合相关性的特征来甄别翻译对是否有效。

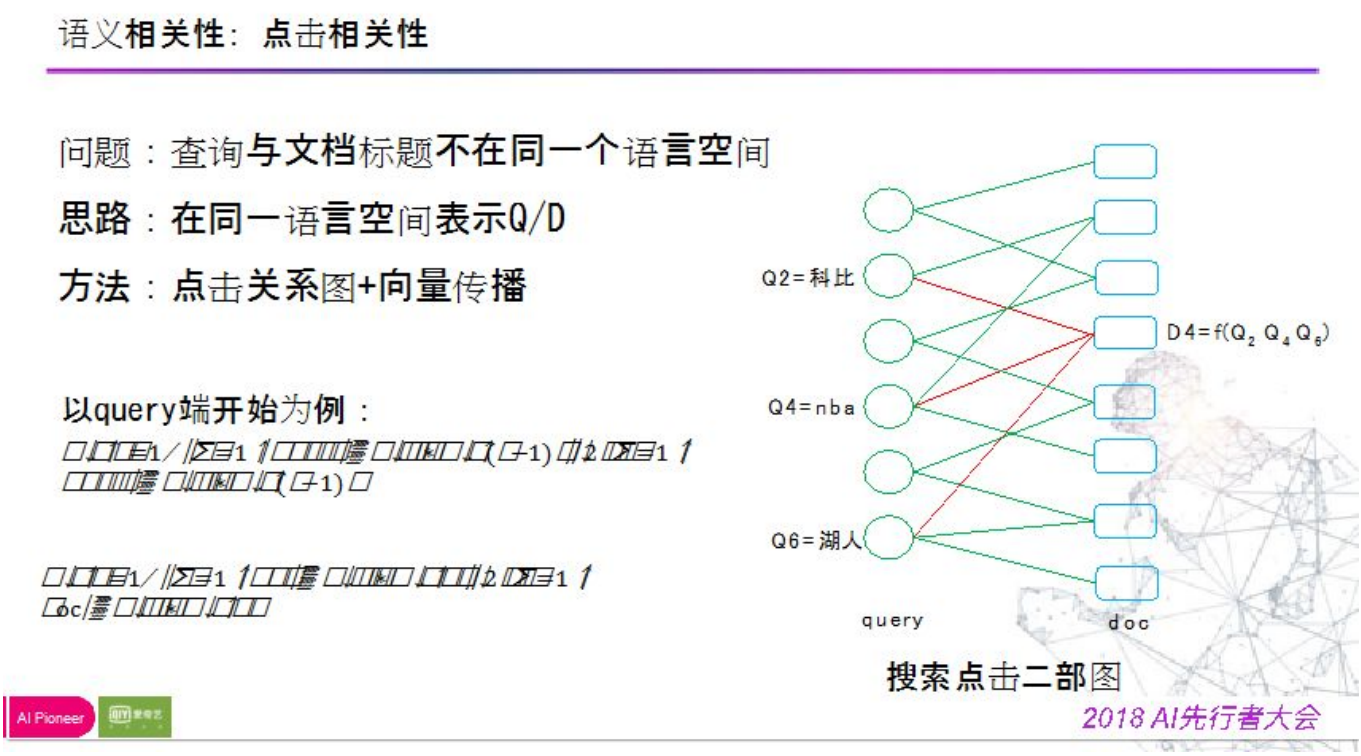
基于这样翻译过程拓展出来的词汇，能够明显拓宽我们召回的范围，这是第一个解决语义相关性的手段。

2.2点击相关性

第二个解决语义相关性的手段是点击相关性。

做一个假设，当用户有一个搜索需求时，假设其用到的搜索词和编辑取的标题不在同一个语义空间，那么该场景造成的mismatch现象会非常严重。那么此时，我们就需要把二者映射到同一语义空间，以提升命中概率。

具体做法：利用搜索点击日志，来构建一个搜索点击二部图。如下图中可以看到：doc4与query2和query4和query6有较强的相关性，虽然此时我们并不知道doc4是什么内容，但是我们已然不难看出，三个query词之间具有较强的相关性，并且这个结论的置信度也是很高的。



这对于查询结果来说是一个不错的拓展方式；在这样的二部图中，我们可以进行多次的迭代，并以次来拓展query的表达。

在构建二部图的时候有许多高挑战性的事情：

> **1)** 图的构建，点击存在噪音，图的内容足够高的置信度；边的权重设计，因为展示数量分布不均，点击数量分布不均，会影响向量传播权；
> **2)** 迭代次数越多，向量传播路径越来越长，泛化能力越强，但是准确率会下降，需要选择最佳迭代次数；
> **3)** 点击关系链接未出现的话，在二部图中是无法出现的，该策略需要用n-gram来拆解拟合，最后用动态规划去选择最优的表达向量。

如下图所示，右侧为一个例子。

2.3 深度学习

第三个解决予以相关性的手段是深度学习，该方式在nlp中应用非常广泛。在搜索场景下，用一些nlp工具，能够把词表示成低维的向量，该向量可以表示词与词之间的相关性，在网络里面加入rnn，cnn等机制，把网络做的足够复杂，以提取更加有效的匹配的特征。同时，我们在文本匹配或者搜索语义匹配的时候，其实要做的就是计算多个文本词序列之间的相关性，我们把词向量和网络结合在一起就可以解决该问题。

在传统的语义文本相关性中有两种计算框架：

> **1) 基于表达**：将文本串通过模型来表示成向量，并用向量相似度来计算文本相似性（如dssm）；
> **2) 基于交互**：在最底层将query和document中的每个词都计算相关性，以此得到相关性矩阵。

如下图所示：

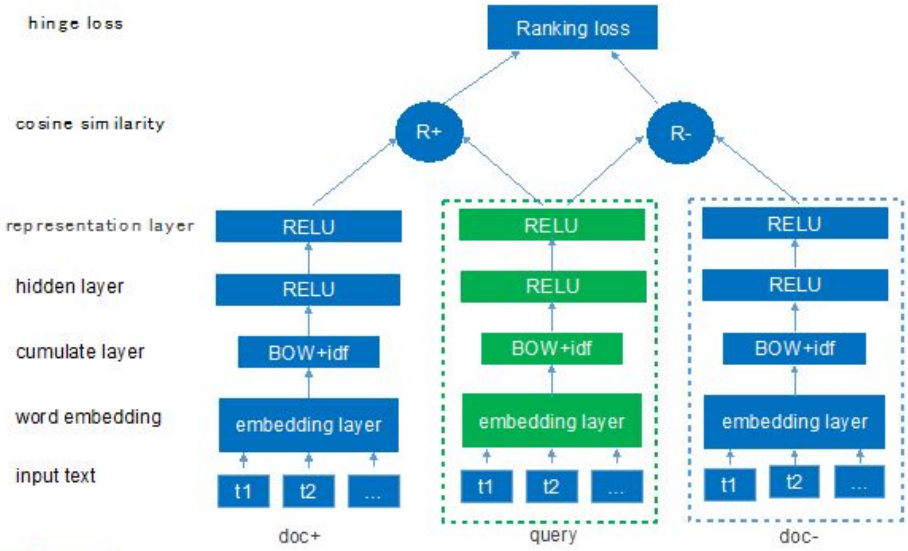
语义相关性：深度语义匹配

- 思路：
  - Word embedding表征词的相关性
  - RNN/CNN/Attention等建模上下文信息
  - 深度网络学习文本序列相关性
- 两种框架：
  - Representation focused
    - DSSM: (Huang et al., CIKM '13)
    - CNN-DSSM: (Shen et al. CIKM '14)
    - LSTM-DSSM: (Palangi et al., TASLP '16)
  - Interaction focused
    - ARC-II: (Hu et al., NIPS '14)
    - MatchPyramid: (Pang et al., AAAI '16)
    - Attention model: (Parikh et al., EMNLP'16)
    - Duet: (Mitra et al., WWW '17)
    - MIX: (Haolan et al., KDD'18)

我们当前的策略是基于表达，如下图所示框架：首先，抽取query下的正负样例；之后，做多粒度切词，用embedding做加权平均，得到文本串的向量表示；再经过两个全连接层生成正样例相关性和负样例相关性；在此基础上，构造损失函数使得正样例大于负样例相关性，用反向传播来优化网络参数。

在视频短文本场景下：表达型方式比交互型方式效果好；网络结构和权重对结果影响很大，idf权重很高；最难点在于ground true构建，严重影响语义模型的效果。

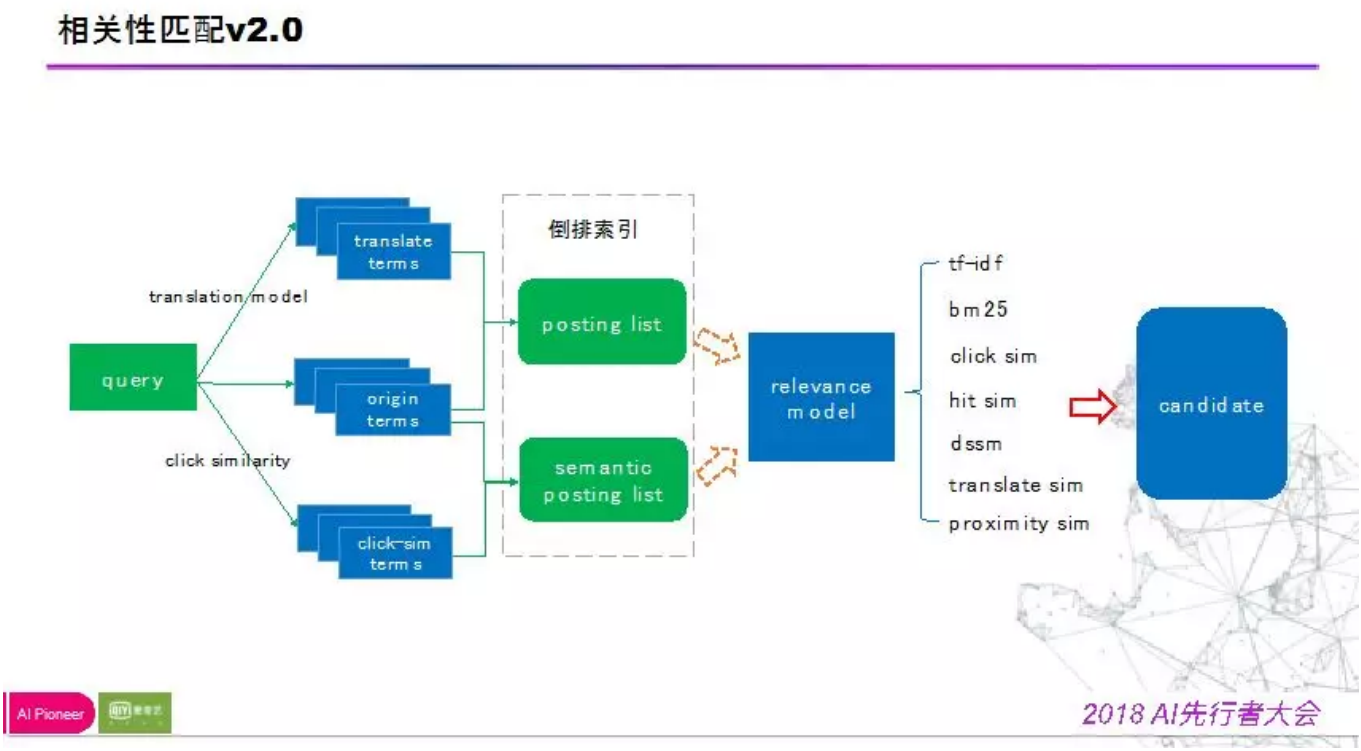
语义相关性：深度语义模型



实验：

- 表示型 vs 交互型
- 网络结构 & 权重
- Ground truth

下图是精确匹配2.0的版本，在这个版本我们基于翻译模型把query进行查询词拓展，同时click-simi的方式去拓展点击相关性的查询词，然后去搜索原倒排索引和语义倒排索引，最后基于相关性模型去计算query和视频内容是否相关。



以上是在解决基础相关性的bad case的时候，应用的语义相关性的技术，这些技术是学术界提出的，在工业界通过a/b test，不断的尝试后得出的比较成功的案例。

四、 排序策略迭代

接下来，我们要介绍的是，在召回了许多跟用户相关的视频之后，面临的排序问题。其实排序问题也有一个逐渐演进的路径：策略排序，学习排序，深度学习模型。

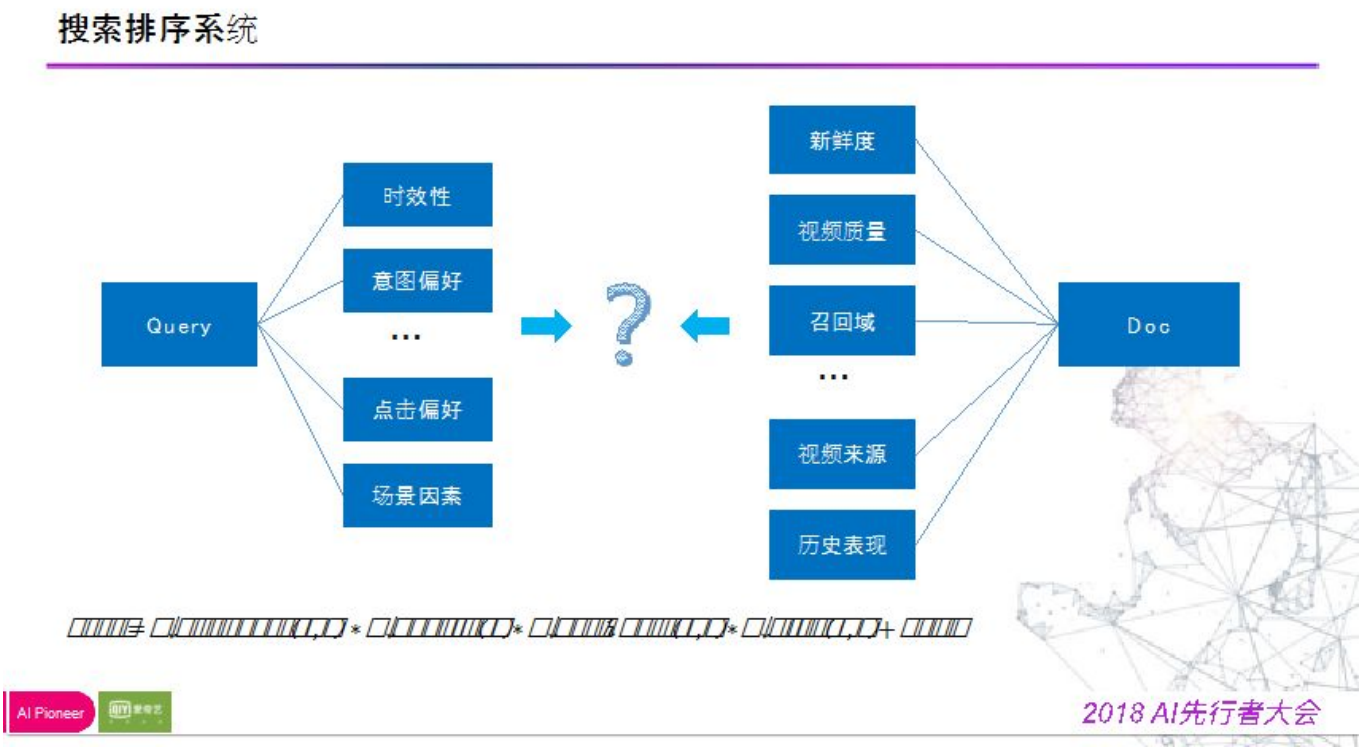
搜索排序面临问题如下：

用户query的时效性（新闻资讯&老电影，游戏&电视剧）；
query场景（新鲜度、语义召回、视频来源、历史点击表现等）。

综合考虑之后，做了一个最初的基于策略的版本，确定了用户的关注要点：

> **1) 相关性**；
> **2) 质量度**，质量更好的结果排在前面如时效性；
> **3) 时效性**，视频从上传开始，其相关性随着时间不断衰减；
> **4) 点击行为**。

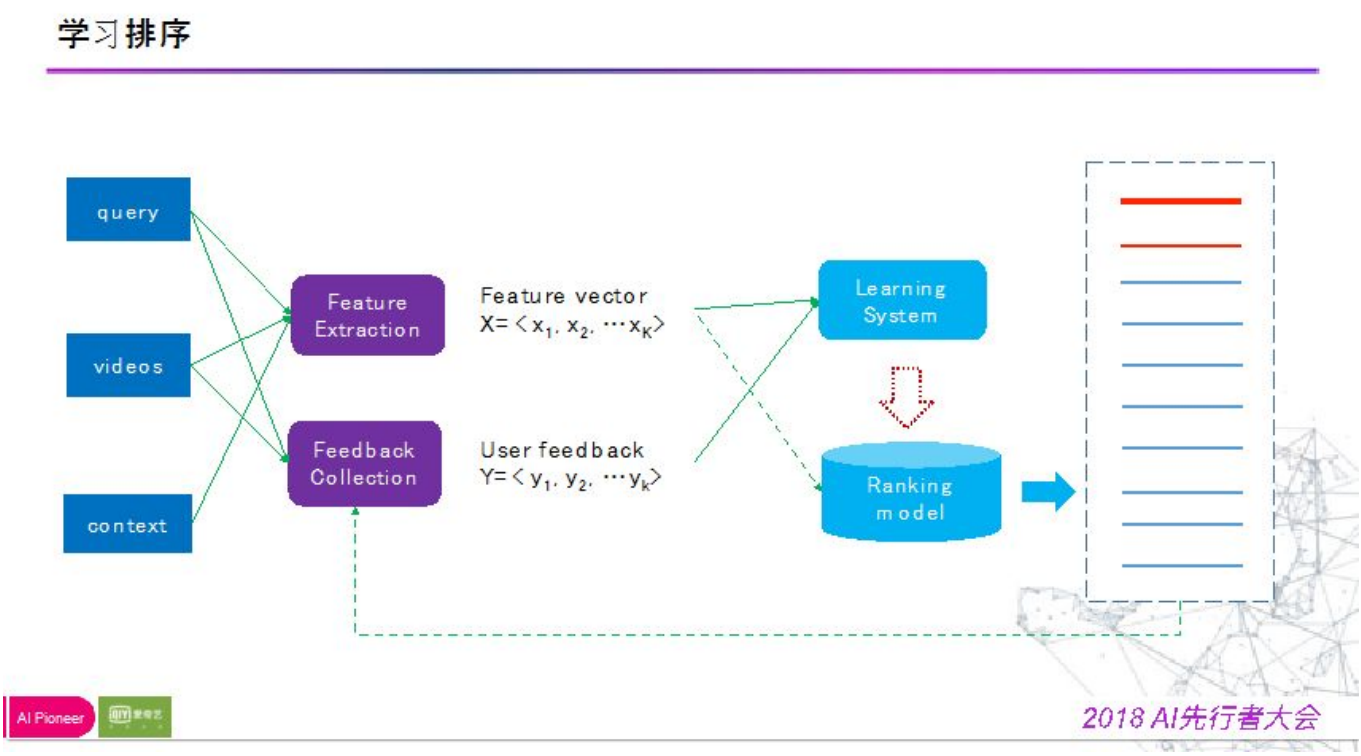
四种因素组合加上产品策略以及规则返回给系统，该版本可以解决大多数常见问题。



后面当我们的特征越来越多的时候，以上四个因素已经开始很难全面地涵盖各种排序因子了。所以，我们需要了解：策略系统无法得到最优解，因为需要不断根据经验去尝试参数。

所以后面迁移到了学习系统，基于用户在历史的排序结果的点击行为收集起来构造label，根据用户在搜索时候给出的query以及展现给用户的video以及上下文信息构造特征向量，与label进行join，得到ground turth，之后进入学习系统进行学习，训练处一个排序模型，就可以对数据进行预测排序。

下图是排序系统的整体流程：



对于一个排序系统，挑战来自于四点：

> **1) 优化目标**：
> point wise，相关不相关；
> pair wise，A优于B；
> list wise，使得排序效果最优化。
> 由于不同的损失函数反应的目标要求的严格程度，会影响最终的模型从样本中学习的程度；
> **2) 样本**：如何构建正负样本，正负样本比例以及权重对模型会产生要大影响；
> **3) 特征**：先颜特征，后验特征，高维稀疏特征；
> **4) 模型**：学习能力，泛化能力。

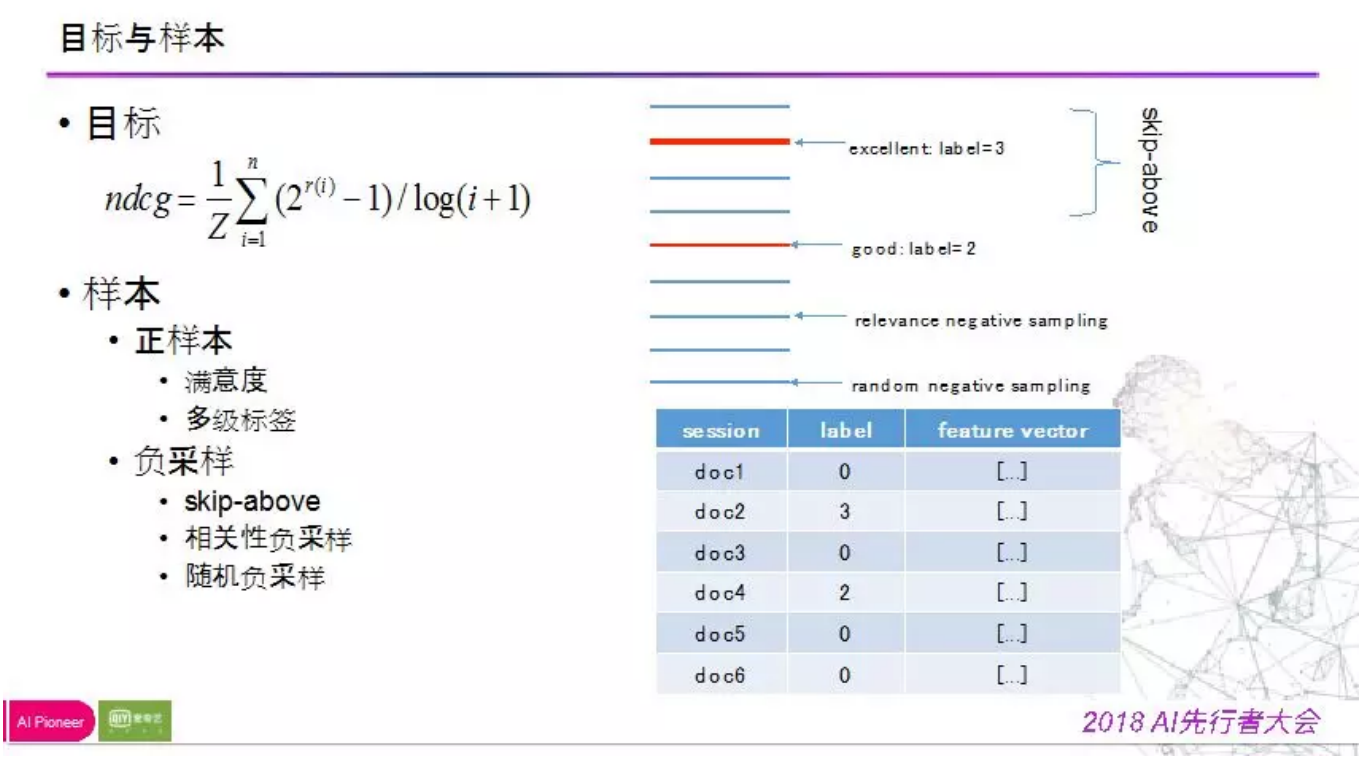
在我们的场景中，

1) 对于目标：

我们最开始选择的是list wise方法。我们采用的优化指标是ndcg，这在搜索引擎中是应用的非常广泛的评价指标。它包含两个参数：
r(i)代表第i个结果的相关性，
i代表i个结果的排序位置。
直观理解：i越小，r(i)越大，ndcg越大，越靠前的结果约相关，这个指标就越高。
这个和搜索引擎的优化目标非常贴近，因此选择这样的list wise学习的优化目标。

2) 对于样本：

用户的点击行为，点击并不代表喜欢，点击后的行为也需要考虑进来，如：点击后观看了多长时间，观看时间占整个视频时间多少，观看市场分布如何，最后会将其映射称观看满意度，量化为三个等级：excllent, good, normal
负样例：skip-above看到的没有点为负样例，相关性负采样，排序靠后的位置做随机负采样，从而构建学习样本。



3) 对于特征：

与业务结合非常紧密，如何把排序场景描述的非常准确，把固定问题泛化，在向量的维度表达出来，即特征提取。Query维度：意图类别（喜欢那个类型的数据），时效性偏好；document维度：质量特征（码流、码率、用户评论、视频帧、视频标签、类别、来源等）；相关性特征：命中特征，bm25等；后验特征：包括用户真实点击率，观看时长，满意程度、点击位置（马太效应影响）、各种维度交叉特征。如下图所示：

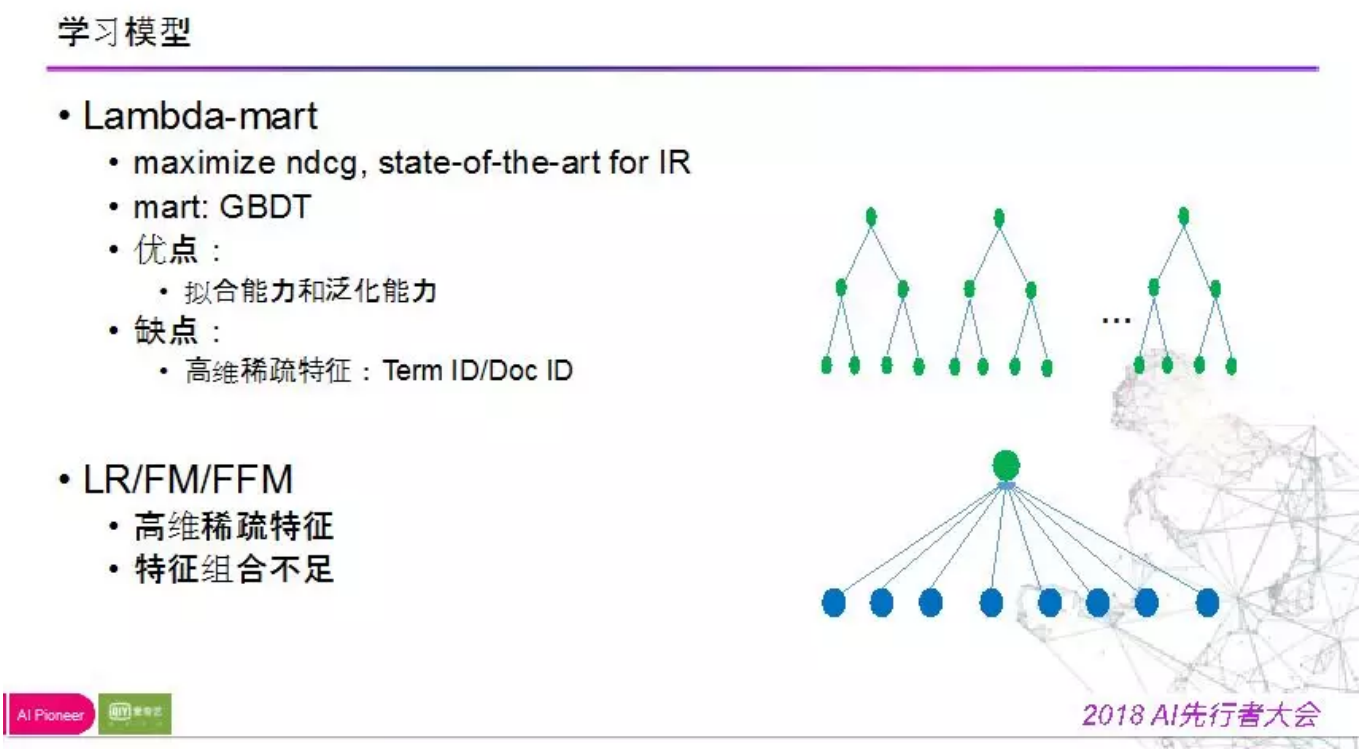
以上是我们在刚迁入机器学习时所采用的特征。

后面我们发现，id特征也是有重要意义的，在特征工程中应该予以考虑，由于我们在提取相关性特征时，是把相关性综合到一个特征中的，该方式丢掉了一些原始信息，那么，如何把这些信息放进去？

另外，query的点击列表，或者说用户的观看列表其实是可以反映出视频的关联信息的，这种信息其实有利于我们做排序优化，我们如何利用这些信息？

所以，第二个版本我们的特征工程中，增加了稀疏的id类特征。

在没有加入稀疏类特征之前，我们的模型是lambda-mart模型，在IR领域是最先进的模型，该模型是一个gbdt模型，基于boosting思想，不断增加决策树，来减小残差。该模型在很多竞赛中表现良好，因为不用过多的特征处理，树模型会考虑特征本身的数据分布，同时有很好的学习泛化能力，树结构很难兼容高维稀疏特征，比方说我们的document是上亿级的特征，很难每个节点走一次树的分割，所以对于加入稀疏特征的时候，树模型会遇到瓶颈。但是在出来高维稀疏特征的时候，像LR、FM、FFM可以认为是线性模型，特征的增加并不会对此类模型造成压力，上亿维也没关系。LR模型弱点在于特征组合能力不足，很多情况下特征组合方式比较重要，树模型从根节点到叶子节点的路径其实是一种组合方式。如下如所示：

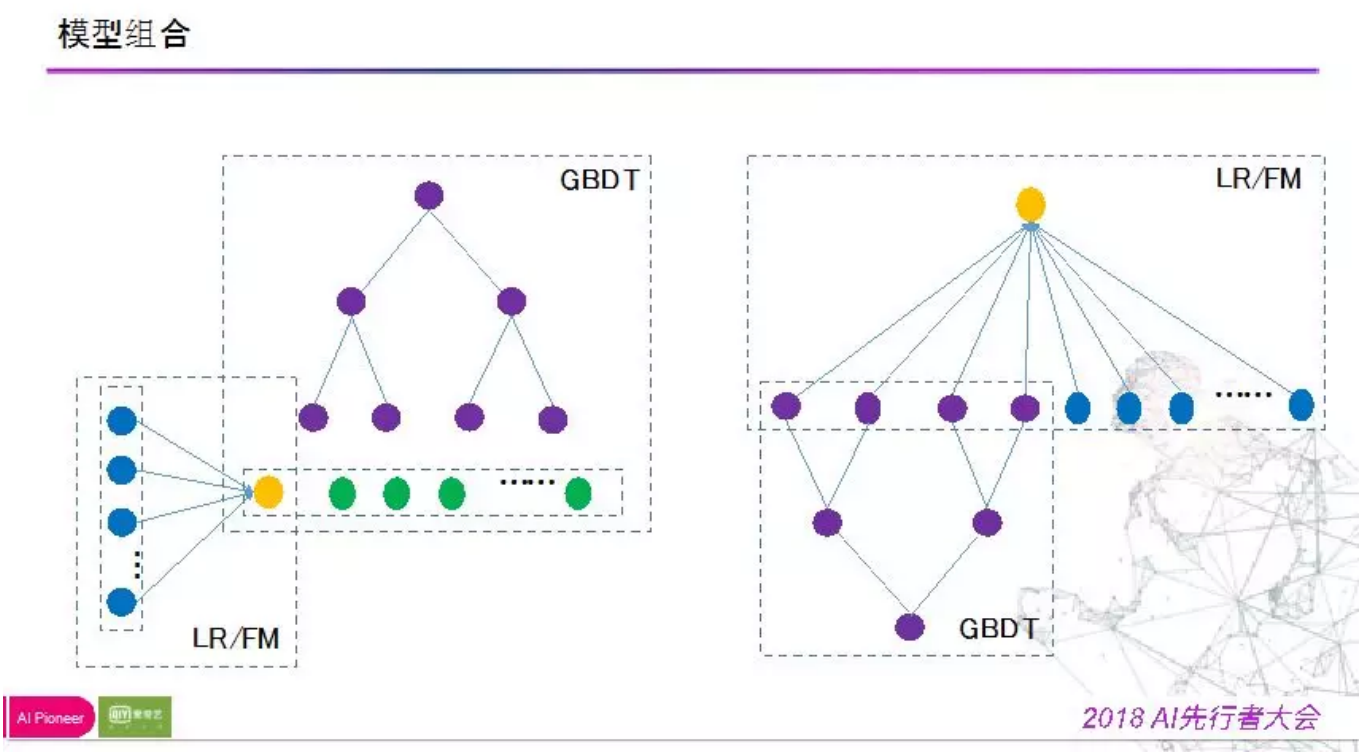


所以针对两类模型的优缺点，我们做了进一步的模型融合的尝试：

> **第一种方式**，用LR模型把高维稀疏特征进行学习，学习出高维特征，把该特征和原始特征做拼接，学习gbdt模型。该方法效果不好，提升很弱。
>
> **分析原因：**把高维特征刚在一个特征去表达，丢掉了原始的特征。

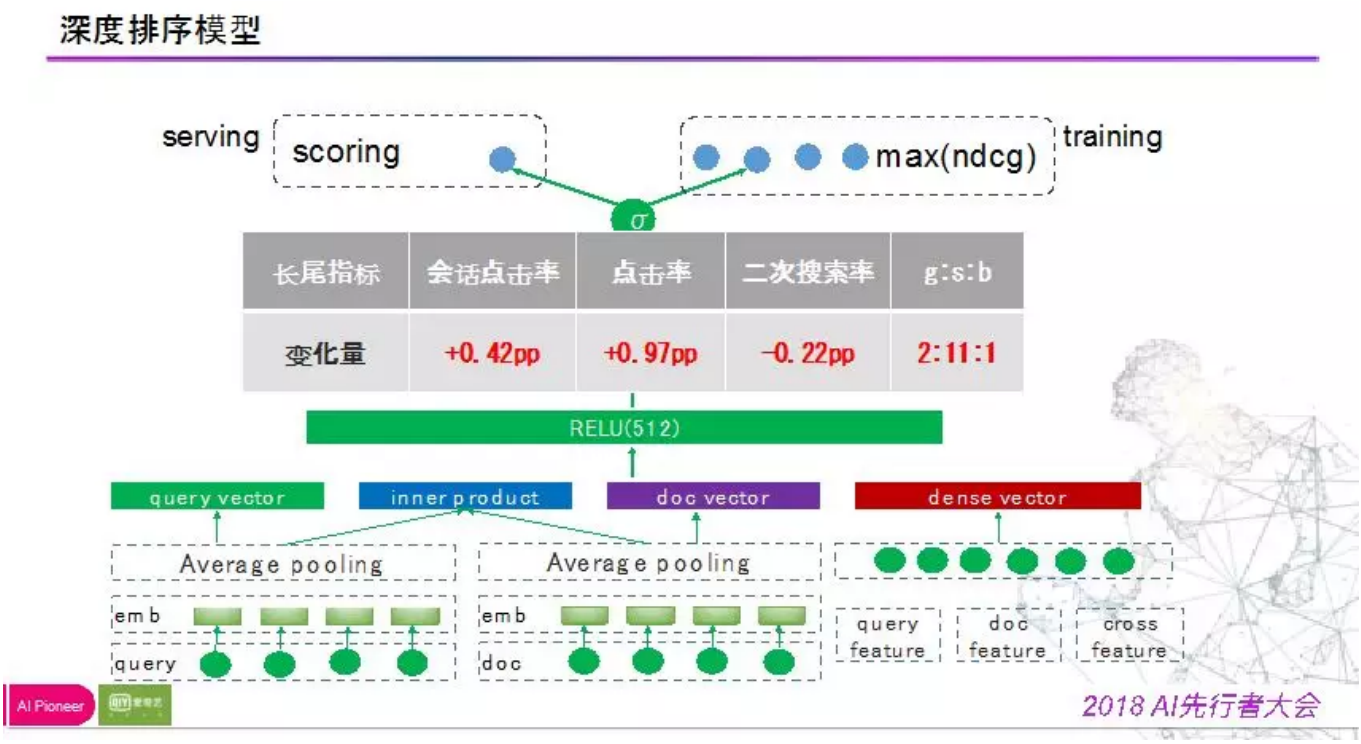
> **第二种方式**，用gbdt去学，学习后把样本落入叶子节点信息来进来与高维稀疏特征拼接，在此基础上用LR学习。
> 该模型效果变差。
> **分析原因：**点击类和交叉类特征是对排序影响最大的特征，这类特征和大量的稀疏类特征做拼接的时候，导致重要性被稀释了，导致模型的学习能力变弱。

两次尝试的模型框架如下图所示：



经过这两种不成功的试验之后，我们引入了基于dnn的排序模型，此模型也需要解决：稀疏特征和稠密特征如何去综合的问题。

如图是我们的dnn排序框架：



底层是query和document的一些描述文本做多粒度切词，之后做embedding然后做加权平均，得到document和query的向量表达，拼接这两组向量，同时再做点积，（两个向量越来越相近，拼接的时候希望上层网络学到两个向量的相似性，需要有足够的样本和正负样例，所以我们自己做了点积），同时用稠密特征，即在gbdt中用到的特征抽取出来，与embedding特征做拼接，最后经过三个全连接层，接sigmoid函数，就可以得到样本的score，并在此基础上用ndcg的衡量标准去计算损失，从而反向优化网络结构。

而在online服务侧，则直接用样本去predict得分。这个模型上线之后，效果非常明显。其中，二次搜索率降低（二次搜索率越低越好，说明用户一次搜中）。

五、 总结

最后做一个总结，我们在做搜索引擎算法迭代的基础上，一直沿着两条路：相关性迭代，怎么去计算更准确，召回更多结果，包括基础相关性，语义匹配以及知识图谱优化相关性计算；同时在排序模型，从集与策略的模型演化到机器学习的模型，后面解决稀疏特征和稠密特征融合的深度学习的排序模型。最后还有做冷启动的时候用到的强化学习的模型，不过时间有限，在此不做详细介绍了。***配套PPT下载，请识别底部二维码关注社区公众号，后台回复【1127】***

作者介绍：

陈英傑，爱奇艺研究员。研究方向：信息检索、机器学习。2012年加入爱奇艺，一直从事搜索排序、搜索用户引导、文本挖掘等工作，参与完成爱奇艺自主研发的搜索引擎，带领rank团队完成从启发式排序策略

到学习排序模型的迭代。依托爱奇艺海量的视频资源库和每天数亿级的用户搜索、观看行为，积极引入、尝试最新的研究成果，迭代搜索排序模型，使得搜索质量和转化率都有显著提高。

编辑介绍:

孙锴，目前就职于一点资讯广告技术部门，任高级算法工程师，负责广告点击率的提升以及商业化算法开发工作。

爱奇艺内推信息:

1. 高级推荐算法研究员，负责首页等多个核心区域的个性化推荐服务，提高内容的分发效率；
2. 高级搜索算法研究员，利用海量用户的搜索点击行为，构建搜索排序模型，优化搜索排序算法。

简历发送至：**chenyingjie@qiyi.com**

——END——

DataFun算法交流群欢迎您的加入，感兴趣的小伙伴欢迎加管理员微信:

文章推荐:

「回顾」搜索引擎从0到1
「回顾」神马搜索技术演进之路
「回顾」外卖推荐算法中有哪些机制与手段?
「回顾」搜索引擎算法体系简介——排序和意图篇

社区介绍:

DataFun定位于最“实用”的数据科学社区，主要形式为线下的深度沙龙、线上的内容整理。希望将工业界专家在各自场景下的实践经验，通过DataFun的平台传播和扩散，对即将或已经开始相关尝试的同学有启发和借鉴。DataFun的愿景是：为大数据、人工智能从业者和爱好者打造一个分享、交流、学习、成长的平台，让数据科学领域的知识和经验更好的传播和落地产生价值。