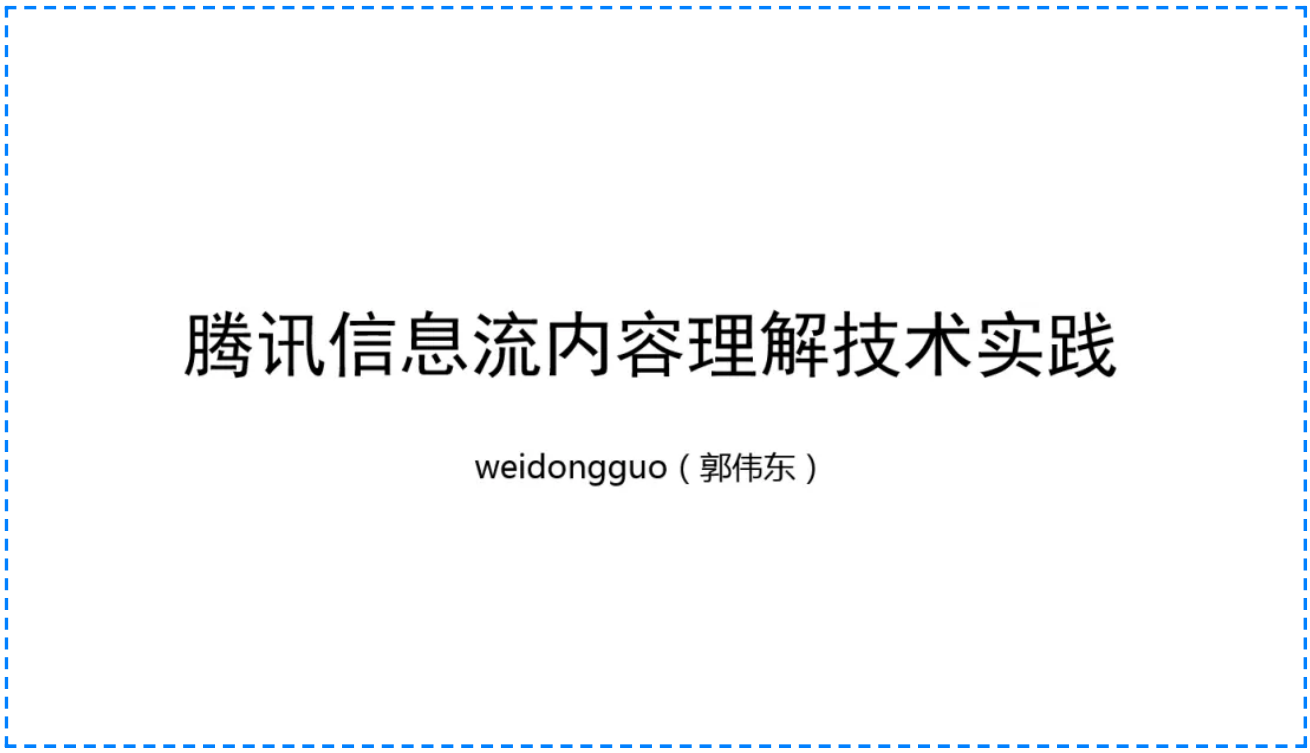
腾讯信息流内容理解技术实践

郭伟东 浅梦的学习笔记 5天前



分享嘉宾：郭伟东

出品社区：DataFunTalk

导读：目前信息流推荐中使用的内容理解技术，主要有两部分构成：1. 门户时代和搜索时代遗留的技术积累：分类、关键词以及知识图谱相关技术；2. 深度学习带来的技术福利：embedding。但是分类对于兴趣点刻画太粗，实体又容易引起推荐多样性问题，而 embedding 技术又面临难以解释的问题。这次主要介绍在信息流推荐中，腾讯是如何做内容理解克服上述问题的。主要包括：

- 项目背景
- 兴趣图谱
- 内容理解
- 线上效果

项目背景

1. 内容理解技术演进

① 门户时代：1995~2002年，主要代表公司：Yahoo、网易、搜狐、腾讯。互联网初期，因为数据较少，因此需要一个内容聚合的地方，人们才能够快速的找到信息。因此，门户通过 **"内容类型"** 对内容进行整理，然后以频道页形式满足用户需求。因为数据少，初期由人工对新闻进行分类。随着数据的增多，靠人工分类已经变得不现实，因此各大公司纷纷引入分类技术，自动化文本分类。此后，文本分类技术发展迅速。

② 搜索/社交时代：2003年~至今，主要代表公司：搜狗、腾讯、Google、百度。随着网络的普及，数据的数量和类型的丰富，门户网站已经不能够承载信息分发的任务。于是，一种新的信息分发技术诞生——搜索。搜索除了需要分类信息以外，还需要精确知道文章是 **"关于什么的"**，关键词技术很好的解决了这个需求，于是也成为那个时期的研究热点。但是关键词技术有一个问题没办法克服：实体歧义问题（如李白，究竟用户是找诗人李白，还是王者荣耀英雄李白）。2012年 Google 提出知识图谱概念，可以用于解决上述的实体歧义问题，实体链指的问题也有了比较大的进展。

③ 智能时代：2012年~至今，主要代表公司：今日头条、出门问问等。使用2012年来作为智能时代的开始，主要是这一年头条成立。头条定义了一种新的信息分发形式——个性化推荐。虽然个性化推荐技术早有研究，但是对于信息分发这个任务有不可或缺的推动作用。

但是在信息推荐中，我们仍然在使用分类、关键词和实体等传统的内容理解方法，那到底在智能时代下是否需要新的内容理解方案呢?

2. 推荐和搜索的区别

推荐和搜索非常相似，都是根据已有的输入，返回跟输入相关的文章，但是对于内容理解的要求区别较大，下面仔细分析下原因：

搜索是给定一个 query 后，预测 doc 被点击的概率进行排序。大致的处理流程如下：首先对 query 分词，得到 < term, weight> 的一个列表（去除停用词等不重要的词），然后根据每一个 term 拉倒排索引 document list 做召回，再对召回的所有文章取并集，最后做整体的排序。注意：这里排序的条件是所有 **term 的交集**（条件概率标红部分）。

推荐是给定一个 user 后，预测 doc 被点击的概率进行排序。大致的处理流程如下：首先查询 user 的用户画像，得到 < term, weight > 的一个兴趣点列表，然后根据每一个 term 拉倒排索引 document list 做召回，再对召回的所有文章取并集，最后做整体的排序。注意：这里排序的条件跟搜索是不同的，排序的条件是 **term 的并集**（条件概率标红部分）。例如用户阅读了王宝强马蓉离婚的新闻，会把 "王宝强"、"马蓉" 作为两个兴趣点积累到用户画像中，而对新的文章排序时候，实际上已经丢失了 "王宝强" 和 "马蓉" 兴趣点是同一篇文章同时积累的这个信息。

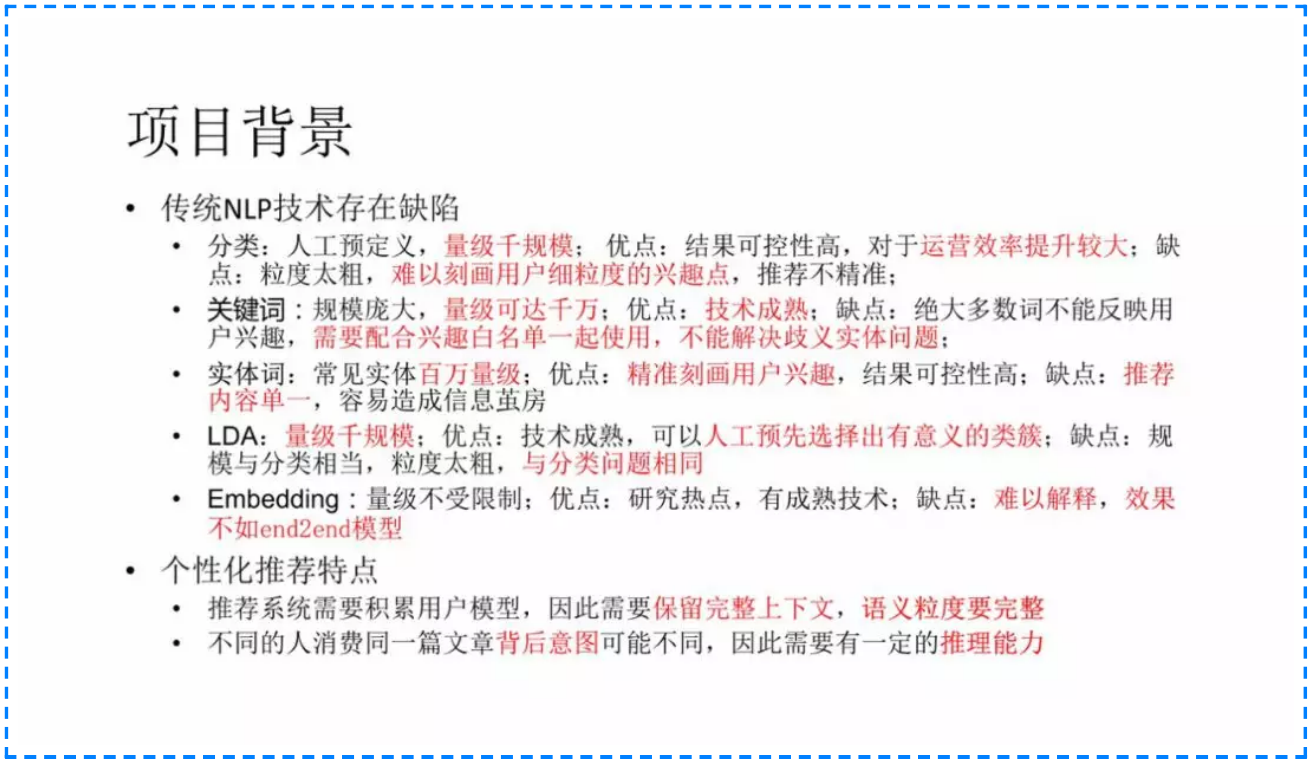
通过上述分析，我们可以得到这样的结论：搜索经过召回之后，排序有完整的上下文信息；但是在推荐中由于经过了用户画像，使用传统的内容理解方案时，排序会丢失用户阅读的上下文

信息。因此，**推荐对于内容理解需要保留完整的上下文**，即把 "王宝强马蓉离婚" 当做一个完整的兴趣点，而不仅仅像搜索一样分别保留 "王宝强" 和 "马蓉"。

3. 用户为什么会消费

传统的内容理解仅仅解决了文章是什么的问题，但是对于用户 "**为什么会消费**" 却没有考虑。举个例子来说，如果我们的一个朋友看了一款 XC60 的汽车，我们会给他推荐什么内容呢？我们一定不会不停的推荐 XC60，而是会猜测朋友可能是喜欢沃尔沃这个品牌，比较看重安全性，或者对于空间有一定要求的 SUV。因此，如果要做好一个推荐系统，内容理解同样也需要相应的能力，挖掘用户真实消费意图。

4. 传统的 NLP 的技术缺陷



以上是整个项目的背景，我们总结一下。传统 NLP 技术存在缺陷：

- 分类：人工预定义，量级千规模；优点：结果可控性高，人工可以参与运营；缺点：粒度太粗，难以刻画用户粒度的兴趣点，推荐不精准；
- 关键词：规模庞大，量级可达千万；优点：技术成熟；缺点：绝大多数词不能反映用户兴趣，需要配合兴趣白名单一起使用，不能解决歧义的问题；
- 实体词：常见实体百万量级；优点：精准刻画用户兴趣，结果可控性高；缺点：推荐内容单一，容易造成信息茧房；
- LDA：量级千规模，优点：技术成熟，可以人工预先选择出有意义的类簇；缺点：规模和分类相当，粒度太粗，与分类问题相同；
- Embedding：量级不受限制；优点：研究热点，有成熟技术；缺点：难以解释。

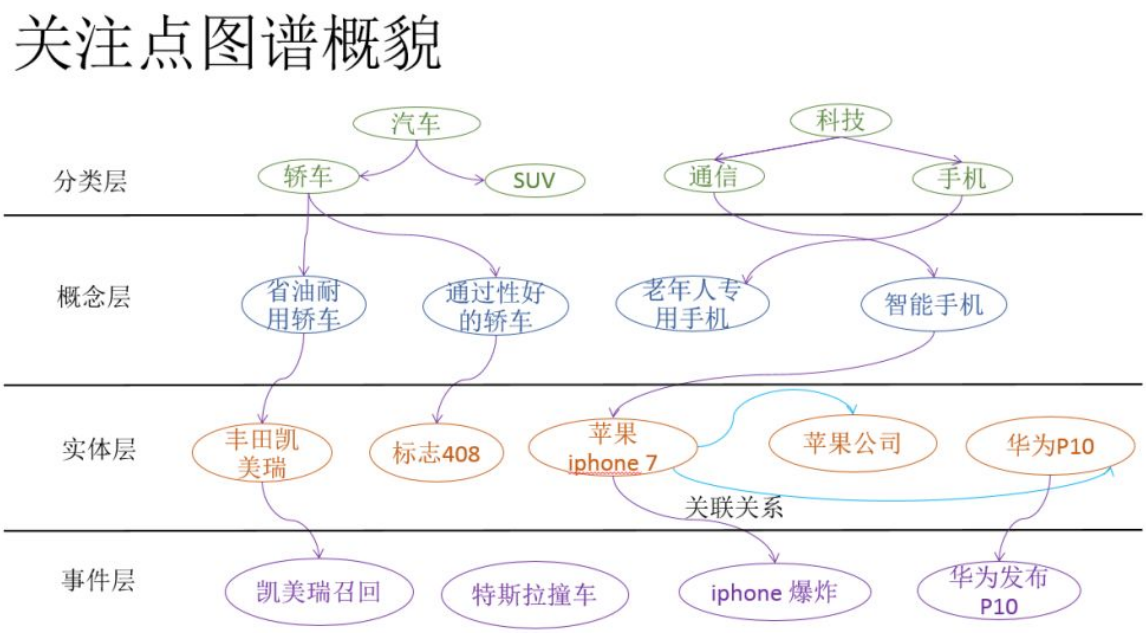
个性化推荐需求：

- 推荐系统需要积累用户模型，因此需要保留完整的上下文，语义粒度要完整；
- 不同的人消费同一篇文章背后原因可能不同，因此需要有一定的推理能力。

因此，传统的内容理解方案并不能很好的满足个性化推荐的需求。个性化推荐不仅需要传统的内容理解方式，还需要一种能够有完整上下文，并且具有推理用户真实消费意图的能力。

兴趣图谱

1. 兴趣点图谱

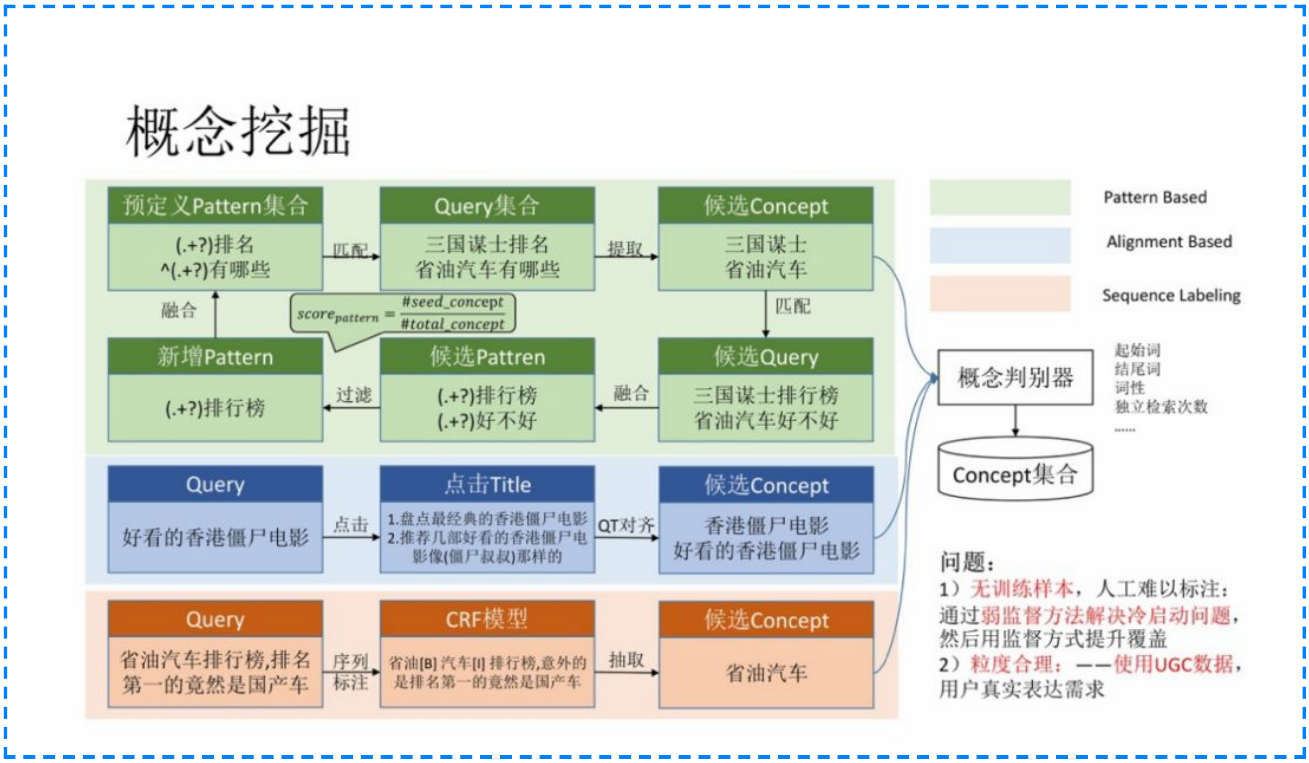


根据上述分析，我们提出了兴趣点图谱，用于解决上述个性化推荐中遇到的问题。兴趣点图谱由四层组成：分别为：分类层、概念层、实体词和事件层。下面分别介绍各层的内容：

- 分类层，一般是由 PM 建设，是一个严格树状的结构，一般在1000左右个节点；
- 概念层：有相同属性的一类实体称之为概念，例如老年人专用手机、省油耐用车等；
- 实体层：知识图谱中的实体，如：刘德华，华为 P10 等；
- 事件层：用来刻画某一个事件，例如：王宝强离婚、三星手机爆炸等。

分类层主要解决人工运营的需求；概念层推理用户消费的真实意图；实体层负责一般兴趣点的召回；事件层精准刻画文章内容。下面介绍如何构造兴趣点图谱。

2. 概念挖掘



概念本质是一种短语，其实短语挖掘的论文非常多，像韩家炜老师团队就有很多相关的论文，但是概念有自己的独特性：

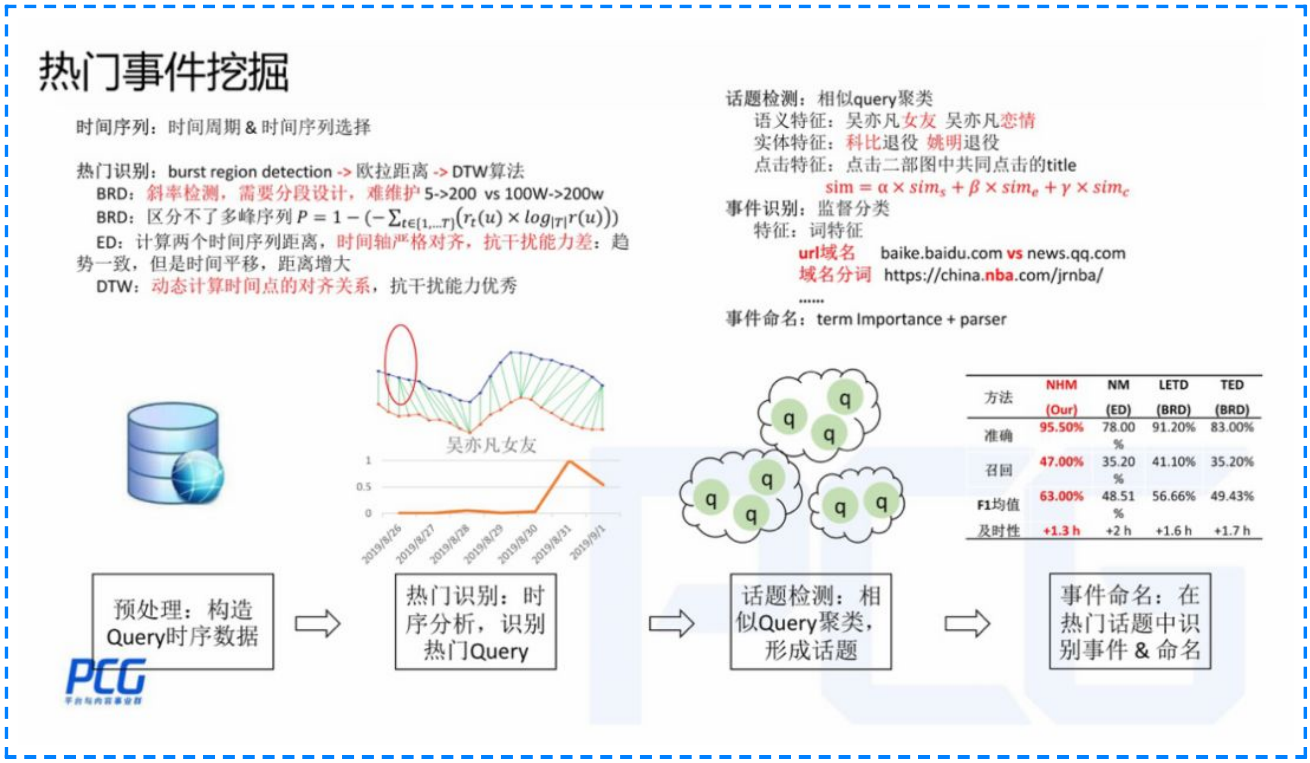
① 没有训练样本，并且人工难以标注。因此只能通过弱监督方法解决冷启动的问题，然后使用监督方式提升覆盖。

② 粒度问题。比如 "明星" 是一个概念，但是太泛，不能精准刻画用户兴趣，但是 "身材好的女明星" 就很合理，那如何描述粒度呢？使用 UGC 数据，用户真实表达需求。

因此，具体挖掘时，我们使用了搜索数据，通过用户的点击行为进行半监督算法的学习。具体算法如上图所示：

挖掘概念使用的是搜索数据，每一个概念都有多个点击的网页，对网页进行实体抽取，然后统计实体和概念的共现频次就可以获得较为准确的上下位关系，我们在 KDD 的 paper 中有详细的介绍，这里就不再重复。

3. 热门事件挖掘



事件指的是热门事件。如果一个事件比较热门，网友就会有了解需求，会通过搜索引擎来查询事件，因此我们使用 query 作为热门事件挖掘的来源。

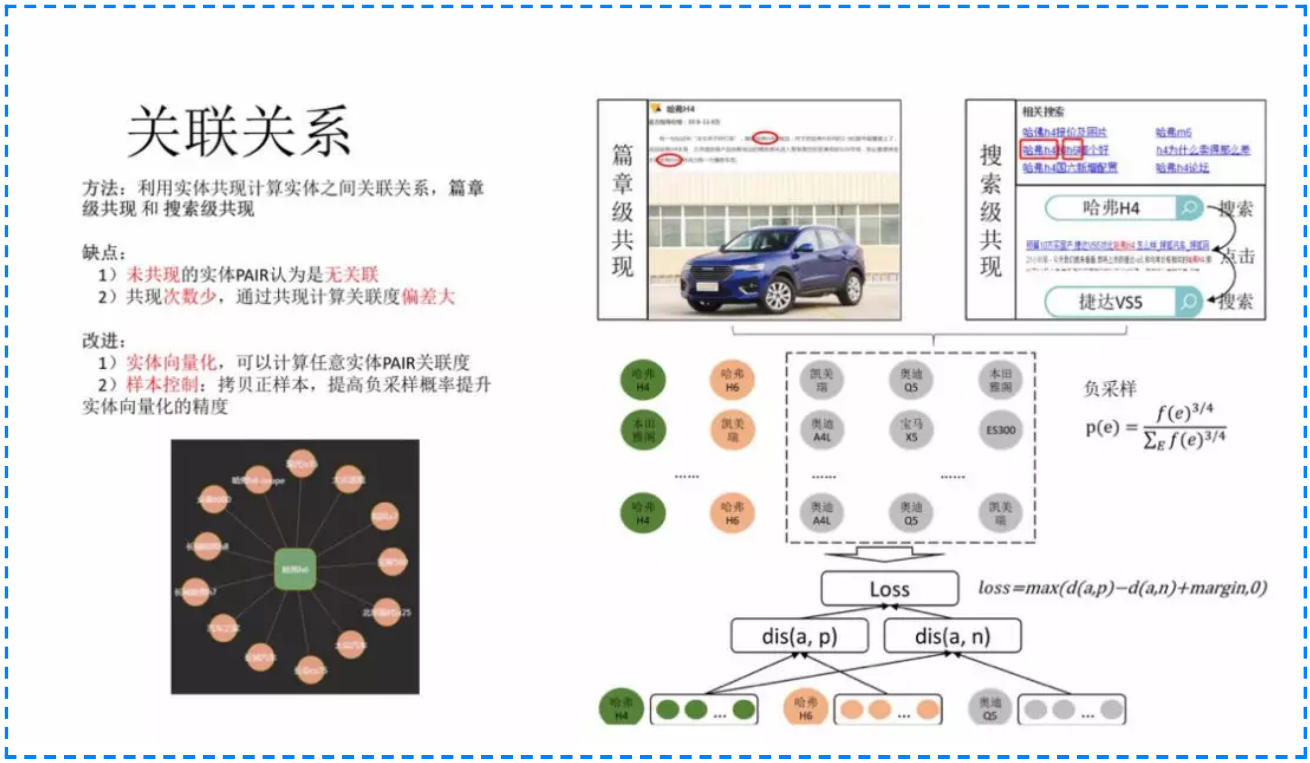
一个比较常见的方法是根据事件搜索量变化趋势判断，常规的做法是 BRD（Burst Region Detection），判断时间序列上是否有爆发点。但是 BRD 会遇到一些归一化，甚至多 point 的问题，于是我们采用了上图的方式克服上述问题：

- 热门识别：时序分析，识别热门 query。首先定义一个热门事件的趋势模板；然后对第一步预处理后的时序数据与热门模板进行相似度计算，如果相似度很高，说明趋势一致，则为热门事件，否则就是非热门。相似度计算的方式最早用的距离是欧拉距离，但是由于欧拉距离需要严格的时序对齐，会造成一些 bad case，因此改用 DTW 算法。

- 话题检测：同一个事件会有多种表述方法，对应多个 query，因此需要把相同事件的 query 聚类到一起，形成话题。

- 事件识别&命名：热门的话题中往往会伴随一些非事件型的话题，如热门美剧更新时，会出现一个热度高潮，上述方法会混入一些非事件，因此我们需要对热门的话题做一个分类。一个非常有效的特征是 url 中的一些单词，会很有区分性。

4. 关联关系



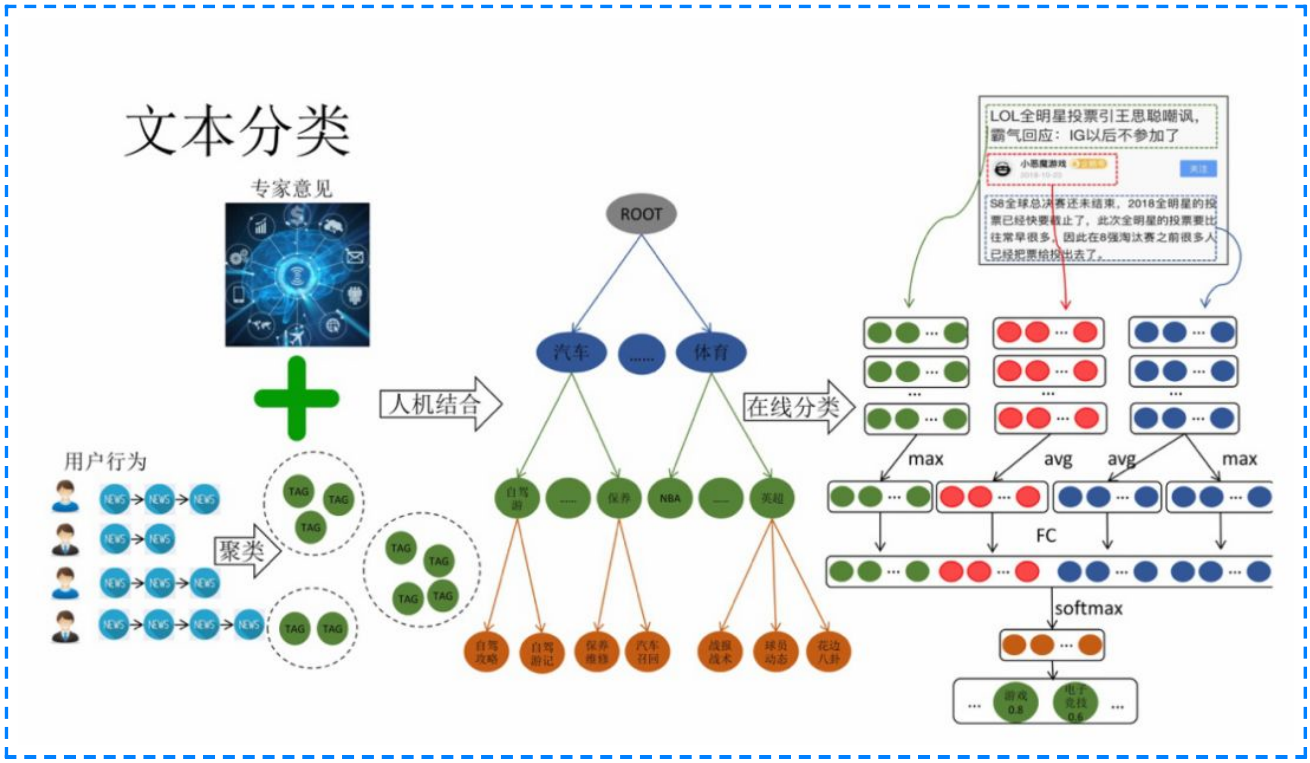
个性化推荐系统中除了要满足用户当前兴趣以外，还需要探索到用户未知的潜在兴趣点，扩展用户阅读视野。因此需要对节点计算关联关系。目前我们仅针对实体做了关联关系的计算。

大家很容易想到，如果两个实体经常会在同一篇文档中出现，应该就是高关联的；或者用户经常连续搜索，即搜完 "刘德华"，然后会马上搜索 "朱丽倩"，应该也是高关联的。确实这种直觉是正确的。虽然这种方法准确率很高，但是会遇到一些问题：没有共现过的，会被认为没有任何的关系；对于共现少的 pair 对，关系的密切度计算误差也会比较大。

因此，需要通过实体向量化的形式克服上述问题。上述的共现数据可以作为正例，负样本采用同类实体随机负采样，正负样本比例1:3，通过 pair wise 的 loss 进行训练，得到每个实体的 embedding，然后计算任意两个实体的关联度。

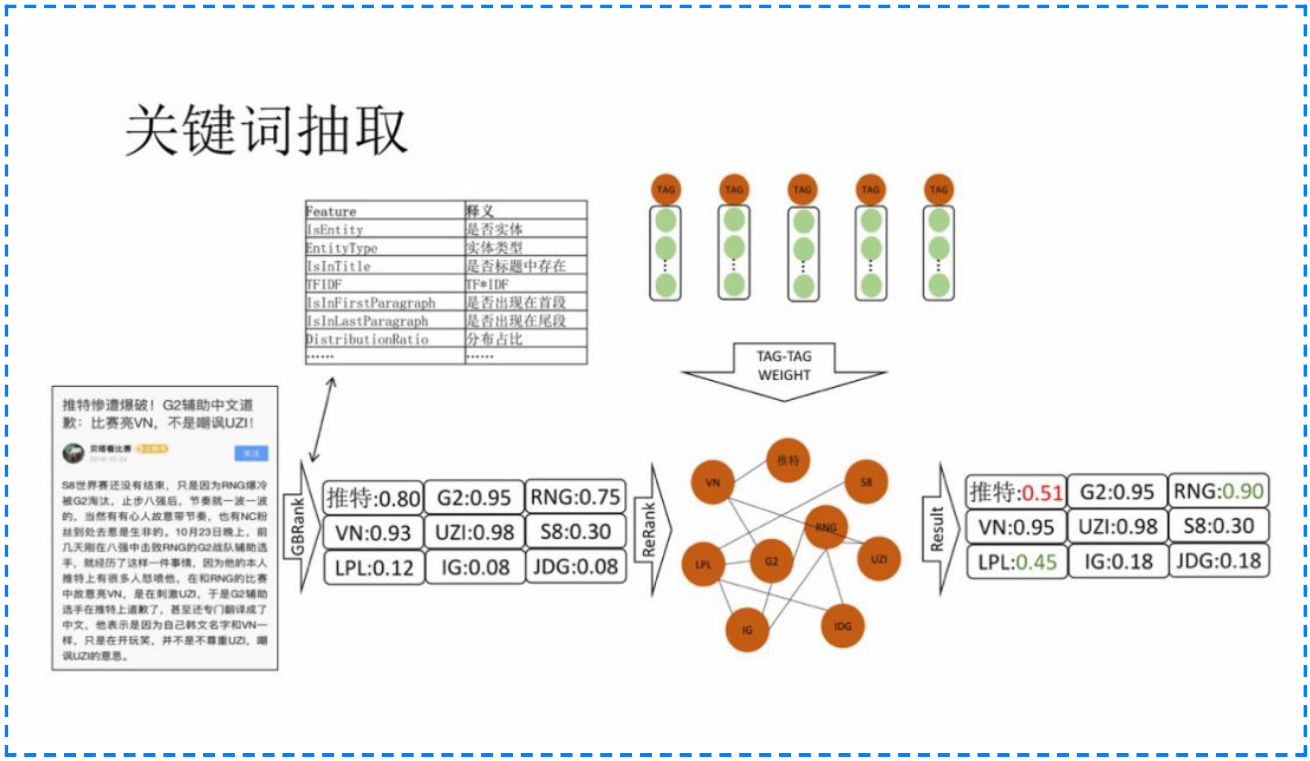
内容理解

1. 文本分类



主题分类层是 PM 整理的，但是 PM 整理的过程中可能会存在一些认知偏差。可以使用用户的点击行为对内容进行聚类，聚完类之后让 PM 去标注，从而总结出一些更适合的类别用于描述用户的兴趣。

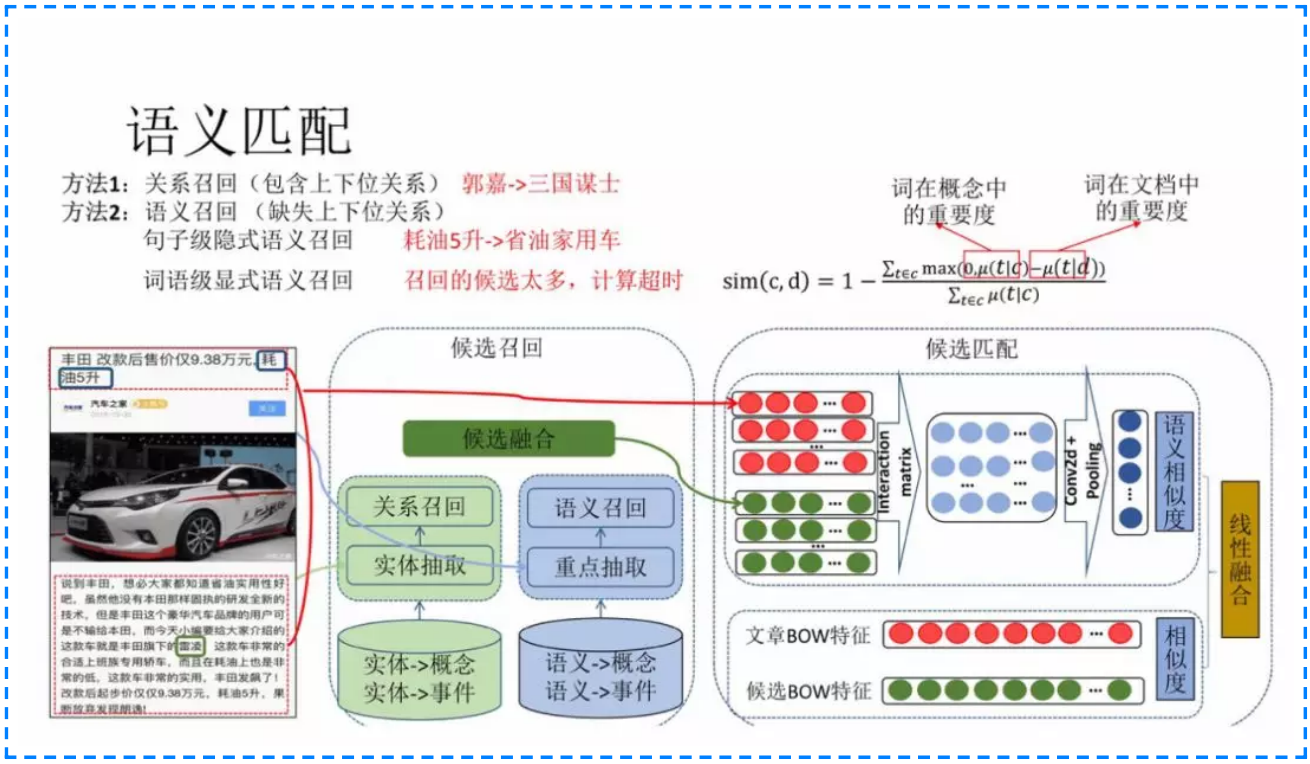
2. 关键词抽取



我们使用了比较传统的关键词提取思路，利用传统特征工程 + GBRank 算法排序。在实际中会遇到这样的问题，如示例，Twitter 出现在 title 中的实体，传统的方式会把 Twitter 分数计算

的很高，但是这篇文章中却不是重点，重点是两支 LOL 战队的骂战。于是我们在 BGRank 之后，加了 re-rank 层，为所有的候选词做一个重排序。词之间边关系使用关联关系 embedding 计算相似度得到。

3. 语义匹配



对于概念和事件类型的标签，原文中可能并不会出现，用抽取的方式就没有办法解决。我们采用召回+排序的方式解决。召回的逻辑分为关系召回和语义召回，其中关系召回会用到兴趣点图谱中的关系数据，召回 1-hop 内的节点作为候选，语义召回通过语义向量召回与 title 近邻的节点作为候选，然后用交互匹配的方式进行排序。如果 1-hop 内的节点数量太多，排序耗时会非常大，因此这里采用粗糙集的方式进行候选的粗排，缩小候选集合再进行排序。

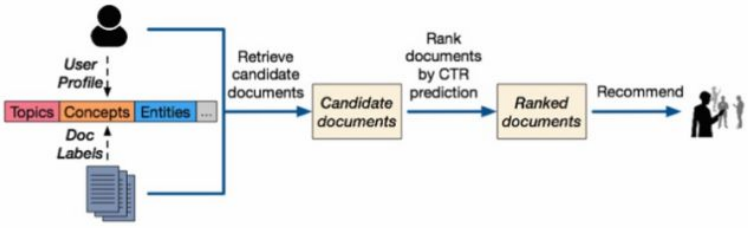
线上效果

线上效果

- **Impression Page View (IPV)**: number of pages that matched with users.
- **Impression User View (IUV)**: number of users who has matched pages.
- **Click Page View (CPV)**: number of pages that users clicked.
- **Click User View (CUV)**: number of users who clicked pages.
- **User Conversion Rate (UCR)**: $\frac{CUV}{IUV}$.
- **Average User Consumption (AUC)**: $\frac{CPV}{CUV}$.
- **Users Duration (UD)**: average time users spend on a page.
- **Impression Efficiency (IE)**: $\frac{CPV}{IUV}$.

Table 4: Online A/B testing results.

Metrics	Percentage Lift	Metrics	Percentage Lift
IPV	+0.69%	UCR	+0.04%
IUV	+0.06%	AUC	+0.21%
CPV	+0.38%	UD	**+0.83%**
CUV	+0.16%	IE	**+6.01%**



实验部分，baseline 是仅用传统的实体和分类标签，而实验组除了实体和分类以外，同时使用概念和事件类型的兴趣点，最后线上效果提升明显。

今天的分享就到这里，谢谢大家。

推荐阅读

- 学习交流小组精彩内容摘要 No.35
- 推荐系统评价：什么是好的推荐系统
- 工业界推荐系统实用分析技巧
- 万字长文解读电商搜索——如何让你买得又快又好

由于微信平台算法改版，公号内容不再以时间序展示，如果大家想第一时间看到我们的推送，强烈建议星标我们和给我们多点点【在看】。星标具体步骤如下图：

（1）点击页面**最上方"浅梦的学习笔记"**，进入公众号主页。

（2）点击**右上角小点点**，在弹出页点“**设为星标**”，就可以啦。

想了解更多关于**推荐系统**的内容，欢迎扫码关注公众号**浅梦的学习笔记**。回复**加群**可以加入我们的交流群一起学习！