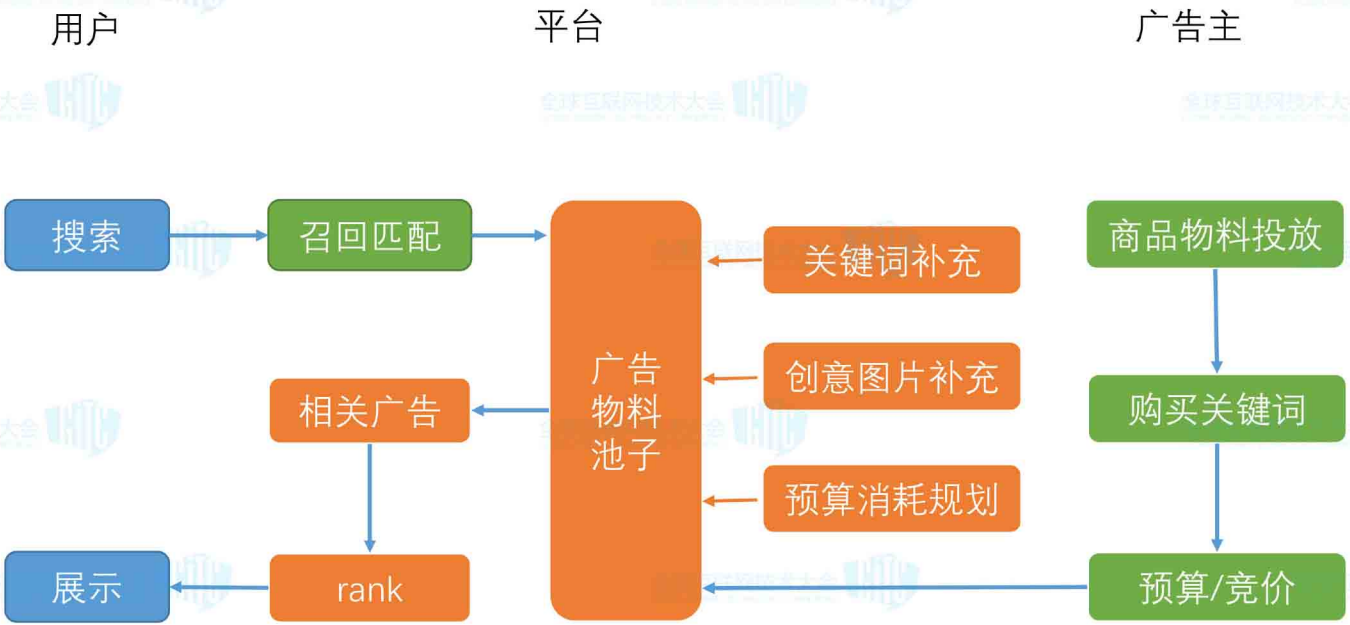
# 电商搜索广告召回匹配

蘑菇街-姜林和

# 内容

- 蘑菇街电商搜索广告形态
- 搜索广告召回匹配和竞价

# 蘑菇街电商搜索广告形态



# 电商搜索广告 vs 搜索引擎广告

| | 电商 | 搜索引擎 |
| --- | --- | --- |
| 搜索词分布 | 流量马太，季节性 | 分布广 |
| 内容 | 商品，图片，描述不足 | 网页 |
| 效果衡量 | 收入，点击/转化率，ecpm | 收入/点击率 |

# 电商搜索广告挑战

Meili Inc.

广告收入

用户体验

商家投资回报

商品丰富度&&召回 && 竞价

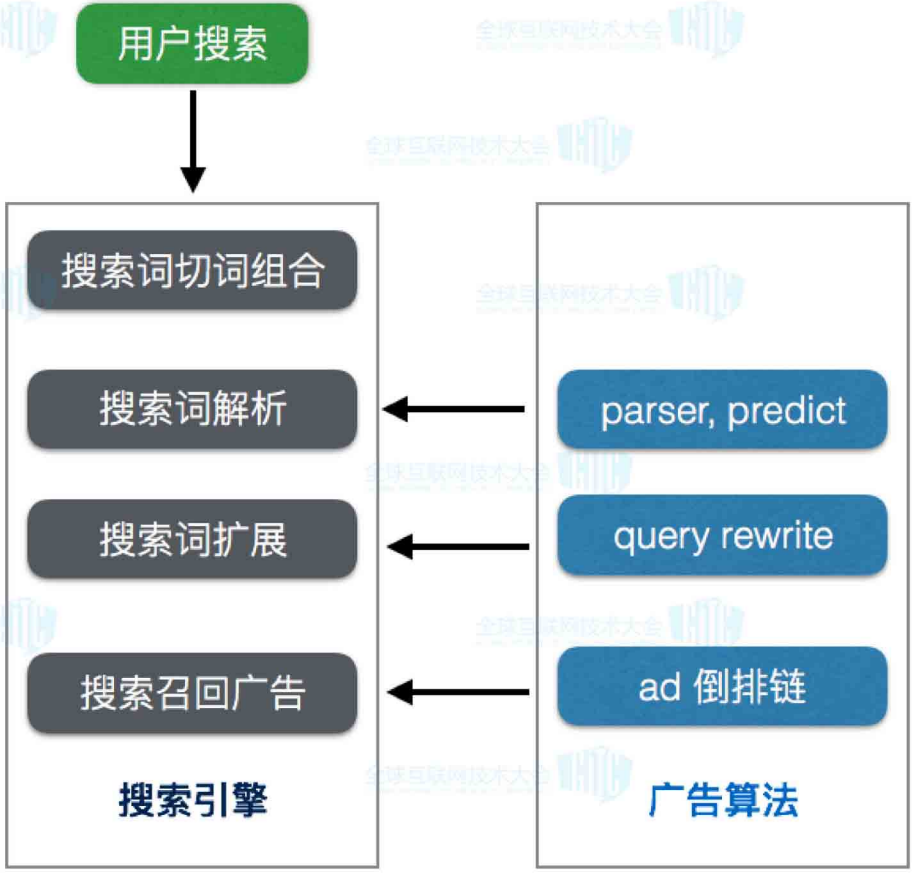
准确

排序优化目标

# 召回匹配

- Query understanding
- Query rewrite
- Ad summary
- Learning to Match

# 广告召回流程


用户搜索
搜索词切词组合
搜索词解析
搜索词扩展
搜索召回广告
搜索引擎
parser, predict
query rewrite
ad 倒排链
广告算法

# Query understanding

目的

对用户搜索词意图的理解

任务

- 解析短语结构, 主要词(sub_query)等。
- 明确用户搜索商品类目(category)

# Query understanding

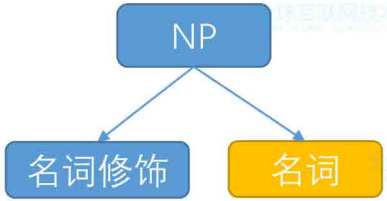
- 电商领域Query 特征
  - 商品类目明确，歧义性少:
    【修身连衣裙】 -> 类目【裙子】
  - 格式固定(名词性短语Noun Phrase)，词顺序不固定
    【修身连衣裙】 <-> 【连衣裙修身】
  - Query 存在季节周期性，流行趋势性

# Query understanding

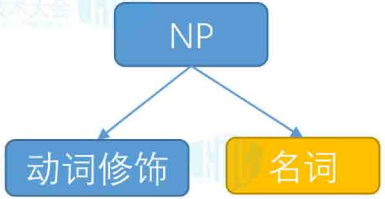
## 名词短语(NP)类型(递归匹配)


连衣裙

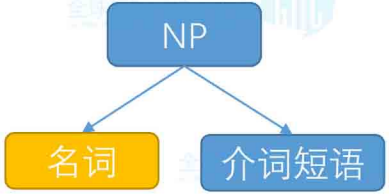

宽松　连衣裙



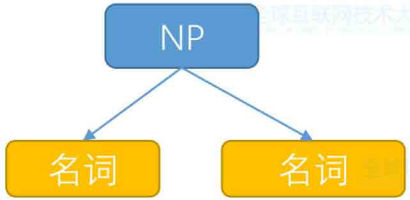

男士　t恤




修身　t恤



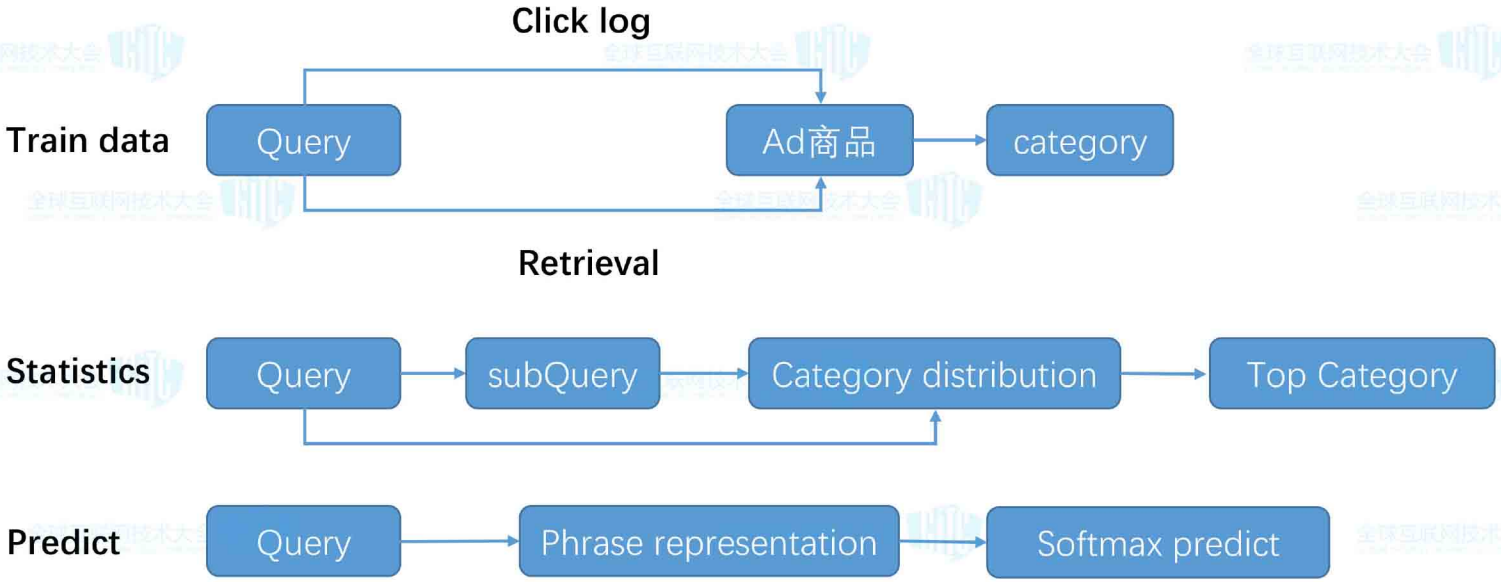

卫衣　带 帽子




耐克　运动鞋

# Query understanding

- 用户搜索词的商品类目预测



# Query understanding

Meili Inc.

## 例子

| query | 类目 |
| --- | --- |
| 裙子 套装 | 套装 |
| 苹果 充电器 | 数码电器 |
| 波西米亚 连衣裙 | 裙子 |

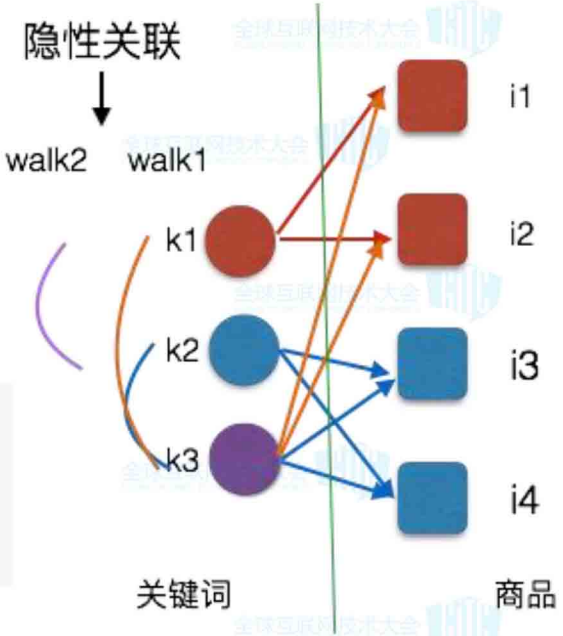
# Query rewrite

- Co-Click
- Co-Session
- Phrase Encoder

# Query rewrite

- 基于用户点击行为

simrank++:用户搜索不同关键词可能会导致的相同商品点击，通过共同的点击商品来建立关键词之间的隐性关联




关键词关联

$$s(q, q') = \frac{C_1}{N(q)N(q')} \sum_{i \in E(q)} \sum_{j \in E(q')} s(i, j)$$

商品关联

$$s(\alpha, \alpha') = \frac{C_2}{N(\alpha)N(\alpha')} \sum_{i \in E(\alpha)} \sum_{j \in E(\alpha')} s(i, j)$$

# Query rewrite

- 基于session
  - 两个搜索词query如果同时出现在同一个用户搜索session里面，且一个词的出现会影响另外一个词出现的概率，那这两个query就存在相关性

- 相关性

  假设$t_1$和$t_2$为数据集中的两个query，以下可以根据统计方式得到
  - 独立: $H_1: p(t_1|t_2) = p = p(t_1|\neg t_2)$
  - 不独立: $H_2: p(t_1|t_2) = p_1 \neq p_2 = p(t_1|\neg t_2)$

# Query rewrite

- 原理

  $t_1$和$t_2$展示的似然函数:

  $$H(p_1, p_2, k_1, k_2, n_1, n_2) = \binom{k_1}{n_1} p_1^{k_1}(1-p_1)^{n_1-k_1}\binom{k_2}{n_2} p_2^{k_2}(1-p_2)^{n_2-k_2}$$

  似然率:

  $$\lambda = \frac{max_p H(p,p,k_1,k_2,n_1,n_2)}{\max_{p_1,p_1} H(p_1,p_2,k_1,k_2,n_1,n_2)}, \quad LLR = -2log\lambda$$

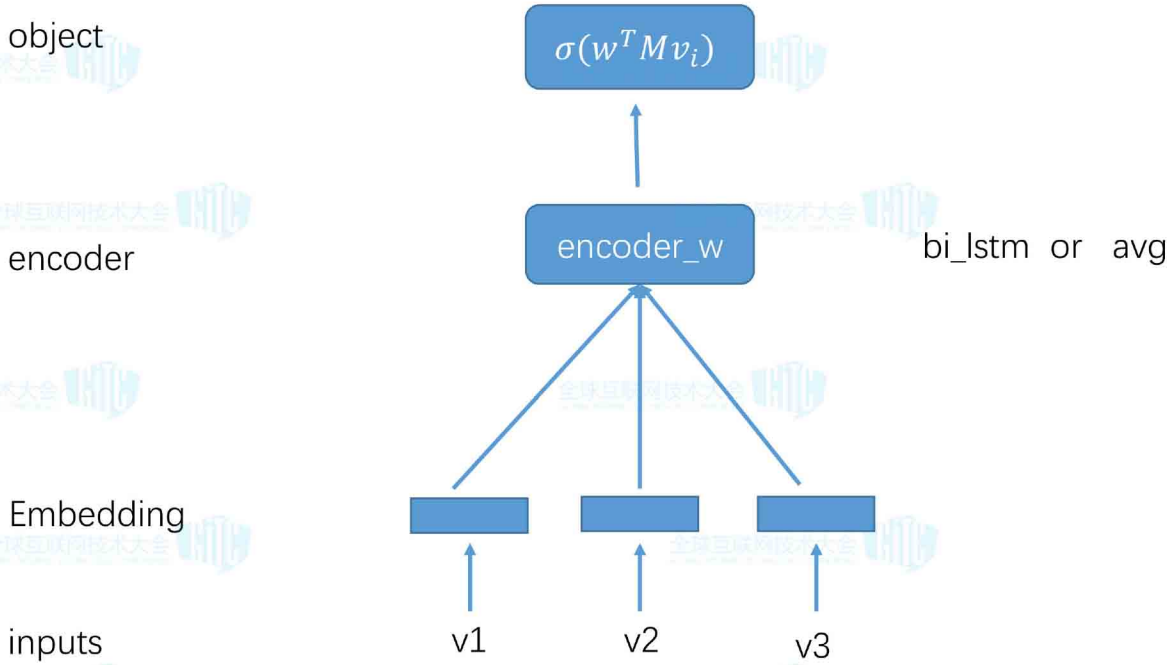
  这里可以求得 $p = \frac{k_1+k_2}{n_1+n_2}$, $p_1 = \frac{k_1}{n_1}$, $p_2 = \frac{k_2}{n_2}$

  $LLR$渐近服从卡方分布

  $LLR$越高，$t_1$和$t_1$得相关性越高，score达到3.85，相关的置信度达到95%

# Query rewrite

- **Embedding -** Phrase Encoder



# Query rewrite

- **Embedding -** Phrase Encoder
  - Pre-train word2vec as 作为Embedding的初始值
  - Category predict: 利用softmax 来预测Phrase的Category
  - Positive: $p(y = 1|\mathrm{sig}moid(w^T M v_i))$ , 其中 w 为 Encoder, $v_i$为短文相关的term的词向量表示
  - Negative: $p(y = 0|sigmoid(w^T M v_s))$ ,其中$v_s$为采样的term的词向量

- **Output**
  - 利用Encoder向量生成Nearest top N

# Ad summary

- **电商广告内容**
  - 图片 + 标题 + 属性

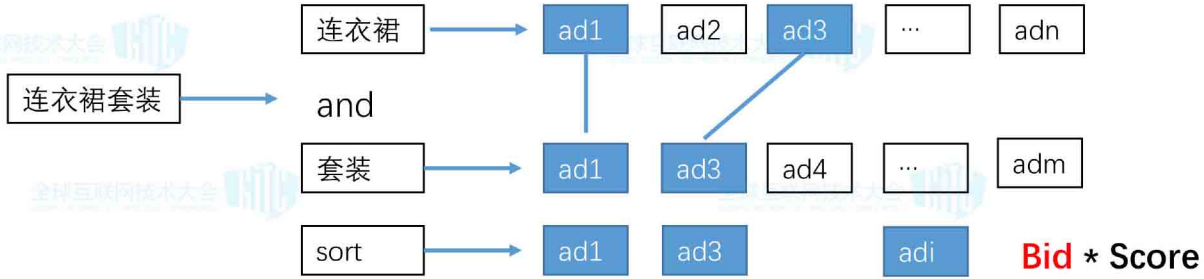
- **如何让用户的搜索快速关联到广告**
  - 建立Ad(广告)到最细粒度term 的索引
  - 建立Ad(广告)到关键词维度的索引

# Ad summary

- **电商广告内容**
  - 图片：基于图片的文本生成，颜色，材质，风格等
  - 标题：切词，停用此，品牌，类目信息
  - 属性：商家填写的属性信息

# Ad summary- Ad 索引

- **建立Ad(广告)到最细粒度term 的索引**




- **方式**
  - ad 不能提前sort, bid(竞价) 非针对全局或者 term
  - 需要两条链全部Merge，然后满足 ad 按 bid * score 排序
- **问题**
  - Rt 高
  - Bid 不能实时更新

# Ad summary- Ad 索引

- **建立Ad(广告)到关键词维度的索引**


**Sort: Bid * Score**

- **关键词**
  - 商品购买的竞价词
  - 常用用户搜索词
- **问题**
  - 如何建立高质量的关键词维度倒排(质和量的权衡)

# Learning to Match

**任务：**用户搜索 query 和 Ad 的Match Score

非端到端:

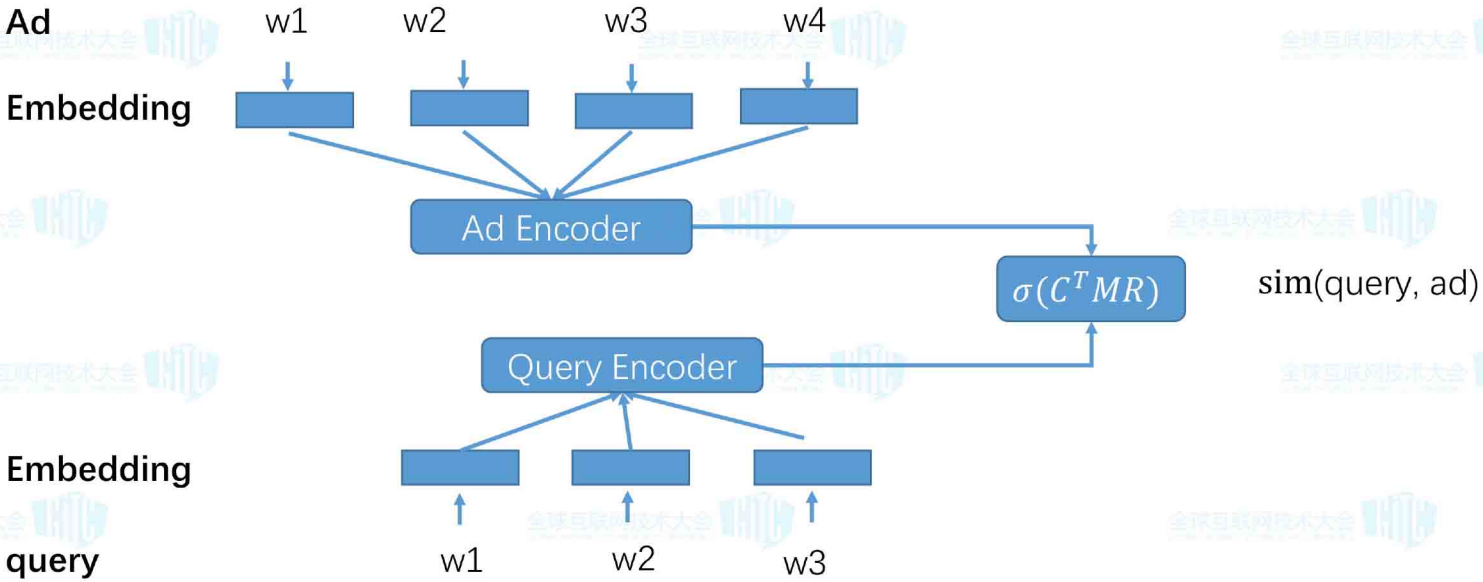
$$\text{sim}(\text{query, ad}) = f(S_{query}, S_{rewrite})$$

$$query_{score} = score(query, subQuery)$$
$$S_{rewrite} = sim(query, rewrite)$$

# Learning to Match

**任务：**用户搜索 query 和 Ad 的Match Score

端到端:



# Learning to Match

- 广告出价的影响:
  - sim(query, ad)衡量query 和广告ad的相关性
  - 广告召回需要考虑广告的出价$S_{Ad} = \text{bid} * \text{score}$
  - score(query, ad)=g(sim, $S_{Ad}$)， g通常是线形的

# Learning to Match

- 其他:
  - score(query, ad)是广告更改竞价后排序位置的依据
  - score(query, ad)是引擎召回 top N截断的依据
  - 召回后的广告进行其他业务目标的排序，如: Ecpm 最大化

Meili Inc.

# Q & A

欢迎交流，蘑菇街数据技术公众号