一种元搜索引擎的查询结果处理模型

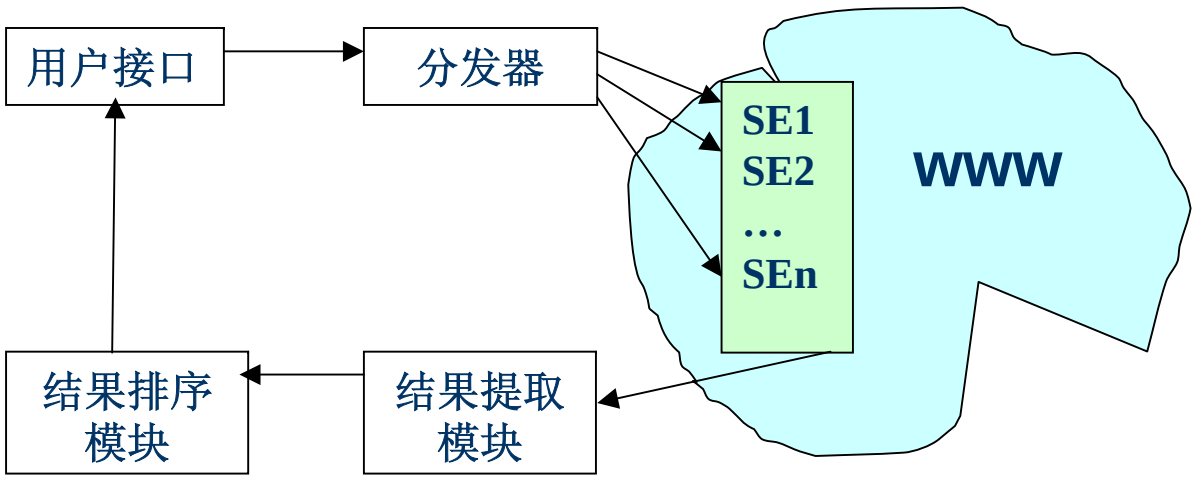
演讲：张强弓

北京工业大学

元搜索引擎

- 调用其它搜索引擎的搜索引擎
- 无需维护庞大的数据库
- 搜索的查全率高

元搜索引擎的体系结构



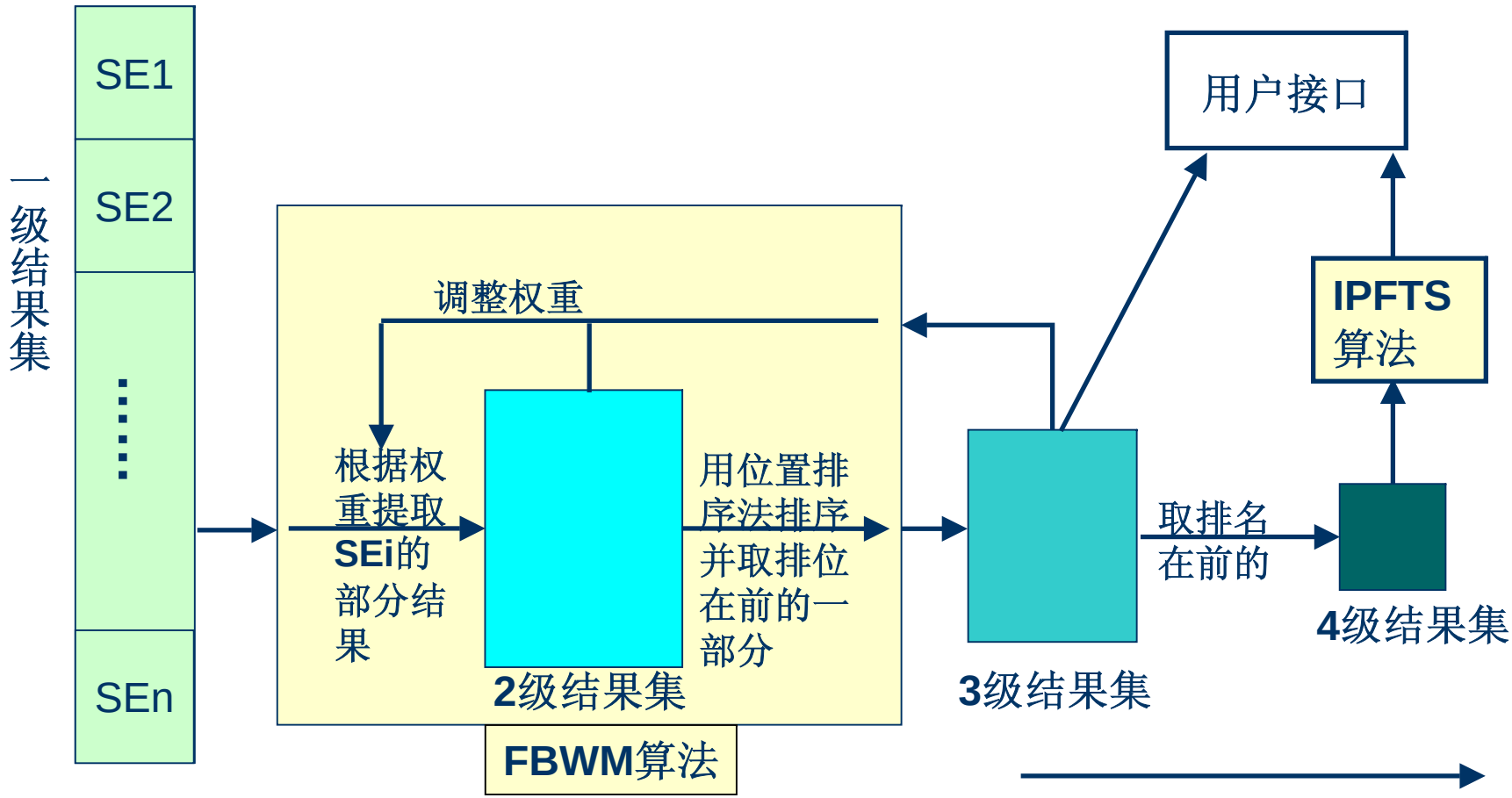
元搜索引擎的体系结构

Refer to **Eric J. Glover's thesis**

本文中的查询结果处理模型

- 指的是查询结果的提取和查询结果的排序
- 关键部分：
  1. 四级结果集结构
  2. 根据反馈调整权重（**FBWM**）
  3. 改进的位置**/**全文排序（**IPFTS**）

模型的工作过程


一级结果集
SE1
SE2
SEn
调整权重
根据权重提取SEi的部分结果
用位置排序法排序并取排位在前的一部分
2级结果集
FBWM算法
3级结果集
取排名在前的
4级结果集
IPFTS算法
用户接口

1. 四级结果集结构
2. 根据反馈调整权重（**FBWM**）
3. 改进的位置***I***/全文排序（**IPFTS**）

1.四级结果集结构的思想

- 元搜索引擎要处理的是非常庞大的结果集
- 我们需要精度更高的算法，但这会耗费更多的时间
- 于是，设置多级结果集，在庞大的较低级结果集中用低级快速的算法挑选出一部分好的结果构成较高级的结果集，在较高级的结果集中应用较高级的算法，形成更高级的结果集。如此往复。
- 目标是保证排名在前的少数结果的准确度。
- 依据是：用户最终可能浏览到的网页少于100个

2.FBWM算法(根据反馈调整权重)

- 结果提取部分
- 结果排序部分
- 调整权重部分

FBWM算法

- 符号的约定：
  - S_i表示元搜索引擎调用的第i个独立搜索引擎，假设共调用M个独立搜索引擎；
  - R_i表示第i个独立搜索引擎返回的结果的结果集；
  - W_i为S_i的权重；
  - N_i表示从R_i中根据W_i提取的结果的数量；
  - n_i表示在3级结果集中，属于R_i中的结果的数量。

FBWM算法的描述

- 结果提取部分

① 对每个独立搜索引擎S_i赋以权重W_0，即$W_i = W_0$

② 将每个R_i中前N_i个结果取出，合并，形成2级结果集，其中

$$N_i = c_1 * |R_i| * w_i \Big/ \sum_{i=1}^{M} w_i$$

$|R_i|$表示集合R_i的基数。C_1是常数，可以取0.1,0.01等。

FBWM算法的描述

- 结果排序部分

③对2级结果集用位置排序法进行排序，取出前n个结果形成3级结果集。其中：

$$n = c_2 * \sum_{i=1}^{M} N_i$$

n的大小由常数C_2 决定

FBWM算法的描述

- 调整权重部分

④ 对每个独立搜索引擎S_i，计算它对3级结果集的贡献比率p_i：

$$p_i = n_i / N_i$$

对p_i作规范化处理，得到贡献比率调节系数P_i

$$P_i = p_i \Big/ \sum_{i=1}^{M} p_i$$

FBWM算法的描述

⑤ 重新调整每个S_i的权重W_i：

$$w_i = c_3 * w_i + c_4 * P_i$$

其中C_3和C_4为常数。
对所有W_i重新计算完毕后，为了保证每次P_i对W_i的影响力是一样的，将每个W_i重新化成百分比形式：

$$w_i = w_i \Big/ \sum_{i=1}^{n} w_i$$

FBWM算法的描述

对每次查询，都重复步骤②到步骤⑤。

FBWM算法的特点

- 对搜索结果总数量的适应性
- 对独立搜索引擎性能变化的适应
- 对独立搜索引擎的搜索精度的差别的适应
  - 用贡献比率而不用贡献率
- 与*训练集方法*的不同之处

3.IPFTS方法 （搜索结果的排序）

改进的位置/全文排序法

IPFTS方法的思想

- 位置/摘要排序[1]法的改进
  - 摘要排序法简单速度快但准确度不高
  - 用全文信息代替摘要信息计算相关度
- 用词条匹配等级使相关度计算更加精确

[1] Refer to 徐宝文，张卫峰《搜索引擎与信息获取技术》

IPFTS方法

- 符号的约定
  - r_i表示第4结果集中第i个结果;
  - D_i表示r_i对应的全部文本信息，称作文档;
  - q表示用户输入的查询串;
  - l_j表示q中第j个词条。
  - X为查询串q中的词条数

词条匹配等级

- 假设一个查询串q有m个词条l_1，l_2，…，l_m，如果在文档D_1中，l_1出现了m次，$l_2,\ldots,l_m$一次也没出现，而在长度与文档D_1相同的文档D_2中，$l_1,\ldots,l_m$各出现一次。显然D_2是比D_1更好的结果。（例如：查询串q=“信息 获取 技术”，则词条“信息”出现很多次而“获取”和“技术”没有出现的文档，与“信息”，“获取”，“技术”都出现的文档相比，自然是后者具有更高的相关度）
- 于是，我们考虑给词条匹配全面的文档奖励：更高的词条匹配等级。

词条匹配等级定义

- 定义：设查询串q有X个词条，文档D中包含有这X个词条的N个（N<=X）,则文档D与查询串q的词条匹配等级为N

IPFTS方法的描述

对第4结果集中的每个结果r_i:

① 仿照摘要排序法，计算文档D_i与查询串q的*普通*相关度（是指没有词条匹配等级影响的相关度）$Rq(q, D_i)$

先计算每个词条与文档的相关度$Rl(l_j, D_i)$

$$Rl(l_j, D_i) = \sum_{k=1}^{Occurence\ (l_j, D_i)} \ln(Length\ (D_i) / Location\ (l_j, k, D_i))$$

再计算查询串与文档的相关度Rq(q, Di)

$$Rq(q, D_i) = \sum_{j}^{X} Rl(l_j, D_i)$$

IPFTS方法的描述

② 计算文档D_i与查询串**q**的词条匹配等级 **MG(q, D_i)**

③ 计算查询串**q**与文档D_i的最终相关度**R(q, D_i)**：

$$R(q, D_i) = MG(q, D_i) * Rq(q, D_i)$$

④ 计算位置信息的得分**P(r_i)**：

$$P(r_i) = 1/i$$

IPFTS方法的描述

⑤ 综合位置信息和相关度信息得到最终排序分数 **Rank(r_i)**

$$Rank(r_i) = c_5 * R(q, D_i) \Big/ \sum_{i=1}^{K} R(q, D_i) + c_6 * P(r_i) \Big/ \sum_{i=1}^{K} P(r_i)$$

其中**C_5**，**C_6**是常数，**K**为第**4**结果集中结果的个数

对第4结果集中的结果按照Rank(r_i)值从大到小排序

IPFTS方法的特点

- 利用全文信息计算相关度能适应各个独立搜索引擎用户接口的差异，同时能保证排名在前的搜索结果的有效性（无死连接）
- 词条匹配等级的引入提高了相关度计算的准确性
- 位置排序法本身的特点

结束语

- 这个模型能减少元搜索引擎对独立搜索引擎的依赖，同时能提高元搜索引擎的查询速度。
- 将来如果能结合用户接口模块和分发器模块的新技术共同应用，将会使元搜索引擎更具威力。