

相关性： 文本匹配

王树森

<http://wangshusen.github.io/>

链路上的相关性模型

- 召回海选
 - 打分量：数万。
 - 模型：文本匹配分数+线性模型，或双塔 BERT 模型。
- 粗排
 - 打分量：数千。
 - 模型：双塔 BERT 模型，或单塔 BERT 模型（交叉）。
- 精排
 - 打分量：数百。
 - 模型：单塔 BERT 模型（交叉）。

文本匹配

- 传统的搜索引擎使用几十种人工设计的文本匹配分数，作为线性模型或树模型的特征，模型预测相关性分数。
- 词匹配分数（TF-IDF、BM25）、词距分数（OkapiTP、BM25TP）。
- 其他分数：类目匹配、核心词匹配等。
- 目前搜索排序普遍放弃文本匹配，改用 BERT 模型；仅剩文本召回使用文本匹配做海选。

词匹配分数

词匹配分数

- 中文分词：将查询词、文档切成多个字符串。
 - 查询词： $q = \text{“好莱坞电影推荐”}$
 - 分词得到： $Q = \{\text{好莱坞, 电影, 推荐}\}$
- Q 中的词在文档 d 中出现次数越多，则 q 与 d 越可能相关。
- TF-IDF 和 BM25 都是基于上述想法。

词匹配分数

Term Frequency (TF)

- 分词结果记作集合 Q ，例如 $Q = \{\text{好莱坞}, \text{电影}, \text{推荐}\}$ 。
- $t \in Q$ 是一个词 (term)，例如 $t = \text{“电影”}$ 。
- 词 t 在文档 d 中出现次数叫做词频，记作 $\text{tf}_{t,d}$ 。
- $\text{tf}_{t,d}$ 越大，说明 t 与 d 越可能相关。
- $\sum_{t \in Q} \text{tf}_{t,d}$ 越大，则 q 与 d 越可能相关。

词匹配分数

Term Frequency (TF)

- 用 $\text{tf}_{t,d}$ 衡量相关性有个缺陷：文档 d 越长，则 $\text{tf}_{t,d}$ 越大。
 - 把文档 d 重复两遍，得到 $d' = d + d$ 。
 - TF 变成的原先两倍： $\text{tf}_{t,d'} = 2 \cdot \text{tf}_{t,d}$ 。
 - 文档 d' 和 d 的信息量相同，算出的相关性分数应当相等。

词匹配分数

Term Frequency (TF)

- 用 $\text{tf}_{t,d}$ 衡量相关性有个缺陷：文档 d 越长，则 $\text{tf}_{t,d}$ 越大。
- 解决方法：用文档 d 的长度（记作 l_d ）对词频做归一化。
- 原先用 $\sum_{t \in Q} \text{tf}_{t,d}$ ，改用 $\sum_{t \in Q} \frac{\text{tf}_{t,d}}{l_d}$ 消除文档长度影响。

词匹配分数

Term Frequency (TF)

- 用 $\sum_{t \in Q} \frac{tf_{t,d}}{l_d}$ 衡量相关性仍然有缺陷：加和同等对待所有 t 。

词匹配分数

Term Frequency (TF)

- 用 $\sum_{t \in Q} \frac{tf_{t,d}}{l_d}$ 衡量相关性仍然有缺陷：加和同等对待所有 t 。
- 词的重要性各不相同，不该同等对待。

词匹配分数

Term Frequency (TF)

- 用 $\sum_{t \in Q} \frac{tf_{t,d}}{l_d}$ 衡量相关性仍然有缺陷：加和同等对待所有 t 。
- 词的重要性各不相同，不该同等对待。如何设定词的权重？
- 语义重要性 (term weight)：电影 > 好莱坞 > 推荐。
 - $t = \text{“电影”}$ 是核心词。
 - $t = \text{“好莱坞”}$ 是重要的限定词。
 - $t = \text{“推荐”}$ 是不重要的词。

词匹配分数

Term Frequency (TF)

- 用 $\sum_{t \in Q} \frac{tf_{t,d}}{l_d}$ 衡量相关性仍然有缺陷：加和同等对待所有 t 。
- 词的重要性各不相同，不该同等对待。如何设定词的权重？
- 语义重要性 (term weight)：电影 > 好莱坞 > 推荐。
- 有多少篇文档包含 t ？好莱坞 < 电影 < 推荐。

词匹配分数


Document Frequency (DF)

- df_t : 词 t 在多少文档中出现过。 (数据集一共有 N 篇文档)
- df_t 介于 0 和 N 之间。
- df_t 大, 说明词 t 判别能力弱, 应当设置较小权重。
 - “你”、“的”、“是” 这样的停用词 (stop word) 的 DF 接近 N , 对判断相关性几乎不起作用。
 - “好莱坞”、“强化学习”、“王者荣耀” 的 DF 都很小, 判别能力强。

词匹配分数

Inverse Document Frequency (IDF)

- Inverse Document Frequency (IDF) 定义为

$$\text{idf}_t = \log \frac{N}{\text{df}_t}.$$


词匹配分数

Inverse Document Frequency (IDF)

- Inverse Document Frequency (IDF) 定义为

$$\text{idf}_t = \log \frac{N}{\text{df}_t}.$$

- IDF 只取决于文档的数据集。
 - 对于人工智能论文数据集，“深度学习”的 IDF 很小。
 - 对于维基百科数据集，“深度学习”的 IDF 很大。

词匹配分数

Inverse Document Frequency (IDF)

- Inverse Document Frequency (IDF) 定义为

$$\text{idf}_t = \log \frac{N}{\text{df}_t}.$$

- IDF 只取决于文档的数据集。
- idf_t 可以衡量词 t 的判别能力； idf_t 越大，词 t 越重要。
- 原本用 $\sum_{t \in Q} \frac{\text{tf}_{t,d}}{l_d}$ 衡量相关性；改用加权和 $\sum_{t \in Q} \frac{\text{tf}_{t,d}}{l_d} \cdot \text{idf}_t$ 。

词匹配分数

Term Frequency—Inverse Document Frequency (TF-IDF)

- 查询词 q 的分词结果记作 Q ，它与文档 d 的相关性可以用 TF-IDF 衡量：

$$\text{TFIDF}(Q, d) = \sum_{t \in Q} \frac{\text{tf}_{t,d}}{l_d} \cdot \text{idf}_t.$$

词匹配分数

Term Frequency—Inverse Document Frequency (TF-IDF)

- 查询词 q 的分词结果记作 Q ，它与文档 d 的相关性可以用 TF-IDF 衡量：

$$\text{TFIDF}(Q, d) = \sum_{t \in Q} \frac{\text{tf}_{t,d}}{l_d} \cdot \text{idf}_t.$$

- TF-IDF 有很多变种，例如：

$$\text{TFIDF}(Q, d) = \sum_{t \in Q} \log(1 + \text{tf}_{t,d}) \cdot \text{idf}_t.$$

词匹配分数

Okapi Best Match 25 (BM25)

- BM25 可以看做 TF-IDF 的一种变体：

$$\sum_{t \in Q} \frac{tf_{t,d} \cdot (k+1)}{tf_{t,d} + k \cdot \left(1 - b + b \cdot \frac{l_d}{\text{mean}(l_d)}\right)} \cdot \ln \left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right).$$

词匹配分数

Okapi Best Match 25 (BM25)

- BM25 可以看做 TF-IDF 的一种变体：

$$\sum_{t \in Q} \frac{\text{tf}_{t,d} \cdot (k+1)}{\text{tf}_{t,d} + k \cdot \left(1 - b + b \cdot \frac{l_d}{\text{mean}(l_d)}\right)} \cdot \ln \left(1 + \frac{N - \text{df}_t + 0.5}{\text{df}_t + 0.5}\right).$$

- k 和 b 是参数，通常设置 $k \in [1.2, 2]$ 和 $b = 0.75$ 。

词匹配分数

词袋模型 (bag of words)

- TF-IDF 和 BM25 隐含了词袋模型假设：只考虑词频，不考虑词的顺序和上下文。
- 例 1：
 - 男朋友 / 送 / 的 / 礼物
 - 送 / 男朋友 / 的 / 礼物

词匹配分数

词袋模型 (bag of words)

- TF-IDF 和 BM25 隐含了词袋模型假设：只考虑词频，不考虑词的顺序和上下文。
- 例 2：
 - 白 / 衬衫 / 灰 / 裤子
 - 灰 / 衬衫 / 白 / 裤子

词匹配分数

词袋模型 (bag of words)

- TF-IDF 和 BM25 隐含了词袋模型假设：只考虑词频，不考虑词的顺序和上下文。
- 词袋模型忽略词序和上下文，不利于准确计算相关性。
- 前深度学习时代有很多词袋模型，例如 Latent Semantic Analysis (LSA)、Latent Dirichlet Allocation (LDA)。
- RNN、BERT、GPT 都不是词袋模型。

词距分数 (Term Proximity)

词距分数

- 查询词 $Q = \{\text{亚马逊}, \text{雨林}\}$
- 文档 $d =$ 我在亚马逊上网购了一本书，介绍东南亚热带雨林的植物群落……
- 虽然 Q 与 d 的文本匹配，但是两者不相关（需求不匹配）。
- 如果用 TF-IDF 或 BM25 计算相关性，会得出错误结论。
- 想要避免这类错误，需要用到词距。
 - 词距： Q 中的两个词出现在文档 d 中，两者间隔多少词。
 - 词距越小， Q 与 d 越可能相关。

OkapiTP

- 词 t 在文档 d 中出现的位置记作集合 $\mathcal{O}(t, d)$ 。
 - t 出现在文档 d 中第 27、84、98 位置上。
 - 那么 $\mathcal{O}(t, d) = \{27, 84, 98\}$ 。
 - 集合 $\mathcal{O}(t, d)$ 的大小等于词频： $|\mathcal{O}(t, d)| = \text{tf}_{t,d}$ 。

OkapiTP

- 词 t 在文档 d 中出现的位置记作集合 $\mathcal{O}(t, d)$ 。
- t 和 t' 是查询词 Q 中的两个词，它们的词距分数：

$$\text{tp}(t, t', d) = \sum_{o \in \mathcal{O}(t, d)} \sum_{o' \in \mathcal{O}(t', d)} \frac{1}{(o - o')^2}.$$

OkapiTP

- 词 t 在文档 d 中出现的位置记作集合 $\mathcal{O}(t, d)$ 。
- t 和 t' 是查询词 Q 中的两个词，它们的词距分数：

$$\text{tp}(t, t', d) = \sum_{o \in \mathcal{O}(t, d)} \sum_{o' \in \mathcal{O}(t', d)} \frac{1}{(o - o')^2}.$$

- 查询词中的 $t, t' \in Q$ 在文档 d 中出现次数越多、距离越近，则 $\text{tp}(t, t', d)$ 越大。

OkαTP

- 词 t 在文档 d 中出现的位置记作集合 $\mathcal{O}(t, d)$ 。
- t 和 t' 是查询词 Q 中的两个词，它们的词距分数：

$$\text{tp}(t, t', d) = \sum_{o \in \mathcal{O}(t, d)} \sum_{o' \in \mathcal{O}(t', d)} \frac{1}{(o - o')^2}.$$

- 查询词中的 $t, t' \in Q$ 在文档 d 中出现次数越多、距离越近，则 $\text{tp}(t, t', d)$ 越大。
- OkαTP 的定义：

$$\sum_{t, t' \in Q, t \neq t'} \frac{\text{tp}(t, t', d) \cdot (k+1)}{\text{tp}(t, t', d) + k \cdot \left(1 - b + b \cdot \frac{l_d}{\text{mean}(l_d)}\right)} \cdot \min(\text{idf}_t, \text{idf}_{t'}).$$

总结

- 词匹配分数包括 TF-IDF、BM25 等。
 - TF：词在文档中出现次数越多越好。
 - IDF：词在较少的文档中出现，则给词较高的权重。
 - 基于词袋模型，只考虑词频，不考虑词序和上下文。

总结

- 词匹配分数包括 TF-IDF、BM25 等。
- 词距分数包括 OkapiTP 等。
 - 查询词 Q 中的词在文档中出现次数越多越好。
 - 查询词 Q 中的任意两个词在文档中越近越好。

总结

- 词匹配分数包括 TF-IDF、BM25 等。
- 词距分数包括 OkapiTP 等。
- 将词匹配、词距等分数作为特征，用线性模型或树模型预测相关性。
- 基于文本匹配的传统方法效果远不如深度学习。

Thank You!

<http://wangshusen.github.io/>