功能

- 支持关键词, 词性, 关键词及词性的任意混合搜索
- 支持严格模式和普通模式. 严格模式下, 所有的关键词必须全部出现
- 支持位置约束, 包括关键词间隔距离和关键词出现的先后位置的约束
- 支持 BM25 与 TF-IDF 两种排序方法, 默认 BM25

展示

任意数目关键词和词性

毫米 加工/v 造纸/v 齿轮 搜索

严格模式 使用TF-IDF

Rank	News	Score
1	0毫米，直径<N>～<N>毫米，齿宽<N>～<N>毫米，最终齿轮精度可达到<N>级甚至更高，加工成本约为磨齿的一半。	10.433589801612614
2	0毫米），加工性能好，是造纸的首选原料。	9.908538354363984
3	0毫米的弧齿锥齿轮。	7.549291933666031
4	0的展品范围将在各类机床、工具与机床附件，金属热加工技术与设备的基础上扩展到制造业广泛需要的自动化控制与动力传动以及其他如轴承，齿轮和模具加工技术及设备等领域。	7.247355474545815
5	0年，“重机牌”数控齿轮加工机床荣获“中国名牌”称号。	7.225061232550399
6	0平方米的展位，精细犹如工艺品、具有奥运元素的“鸟巢”加工吸引了众多人驻足观看，“鸟巢”壁厚只有<N>毫米，加工刀具的直径是<N>毫米。	7.124651794404891
7	0倍模数故意放大，这样加工出来的齿轮在运转时效果还蛮不错的。	7.111778983014283
8	0的滚刀加工齿轮时，就不能以滚刀的外径来选取切削转数。	7.001994217097266
9	0年，含硫、氯的切削油获得专利并应用于重切削、拉削、螺纹和齿轮加工。	6.966148661809973
10	0年，在为客户加工一个设计精度是<N>级的大型齿轮的过程中，徐强创造了奇迹。	6.930668248064804

对比 TF-IDF 与 BM25

BM25

毫米/q 加工/v 搜索

严格模式 使用TF-IDF

Rank	News	Score
1	0平方米的展位，精细犹如工艺品、具有奥运元素的“鸟巢”加工吸引了众多人驻足观看，“鸟巢”壁厚只有<N>毫米，加工刀具的直径是<N>毫米。	7.12521554288073
2	0毫米，直径<N>～<N>毫米，齿宽<N>～<N>毫米，最终齿轮精度可达到<N>级甚至更高，加工成本约为磨齿的一半。	6.618372721820012
3	0是注塑模具精加工得力的工具，该款设备使用的主轴电机功率为4Kw,在合理的工艺方法下，可以使用直径<N>毫米的刀具进行开粗加工，从而保证小刀具精修加工能高效率的完成。	6.489636008368389
4	0毫米），加工性能好，是造纸的首选原料。	5.7811856752628925
5	0毫米，如果不是密封要求极高，那么高的加工精度要求的机械零件有什么用。	5.485227403485496
6	0毫米厚的DDQ钢表层，使复合材料具有与常规钢板相同的外表，但加工性能更优。	5.485227403485496
7	0毫米左右的鹰风筝在风筝的稳定性上，在加工煨制中，制作难易上以及在以后的试飞中，是一种好的方案。	5.192806515920427
8	0毫米为例，初学者可以按照这个尺寸加工制作，但是在选用具体的竹条时，就存在一个竹条刚度差异性，相同截面不同品种和不同部位的竹子其刚度大不一样。	4.975774277863811
9	0亿元，年加工50万吨小麦的面粉加工项目一期工程也将动工。	3.9082932190807895
10	0个加工分厂，一栋研发综合大楼及建筑面积<N>平方米的生产加工现场。	3.9082932190807895

TF-IDF

毫米/q 加工/v 搜索

严格模式 使用TF-IDF

Rank	News	Score
1	0毫米），加工性能好，是造纸的首选原料。	0.8754992101278627
2	0亿元，年加工50万吨小麦的面粉加工项目一期工程也将动工。	0.5955917101768723
3	0个加工分厂，一栋研发综合大楼及建筑面积<N>平方米的生产加工现场。	0.5955917101768723
4	0平方米的展位，精细犹如工艺品、具有奥运元素的“鸟巢”加工吸引了众多人驻足观看，“鸟巢”壁厚只有<N>毫米，加工刀具的直径是<N>毫米。	0.5836661400852419
5	0毫米，直径<N>～<N>毫米，齿宽<N>～<N>毫米，最终齿轮精度可达到<N>级甚至更高，加工成本约为磨齿的一半。	0.5777033550394266
6	0亿元的肉食品加工和保鲜项目合同。	0.5685193597142872
7	0毫米，如果不是密封要求极高，那么高的加工精度要求的机械零件有什么用。	0.5107078725745866
8	0毫米厚的DDQ钢表层，使复合材料具有与常规钢板相同的外表，但加工性能更优。	0.5107078725745866
9	0是注塑模具精加工得力的工具，该款设备使用的主轴电机功率为4Kw,在合理的工艺方法下，可以使用直径<N>毫米的刀具进行开粗加工，从而保证小刀具精修加工能高效率的完成。	0.48557676187263527
10	0万员工构成了欧洲最大的塑料加工市场。	0.4810548428351662

严格模式

全部匹配

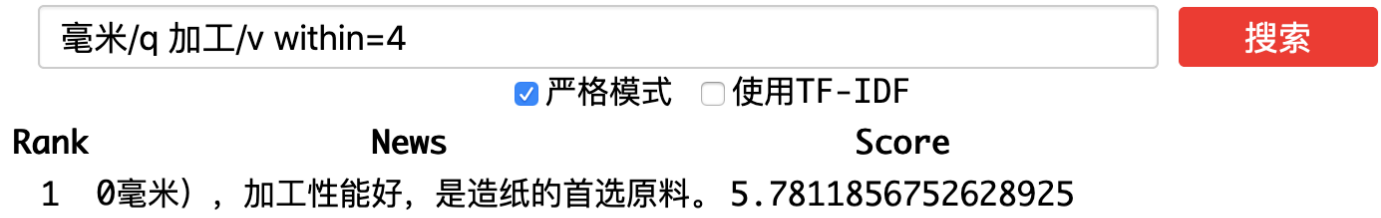
Rank	News	Score
1	0毫米），加工性能好，是造纸的首选原料。	9.906619614690143

位置约束

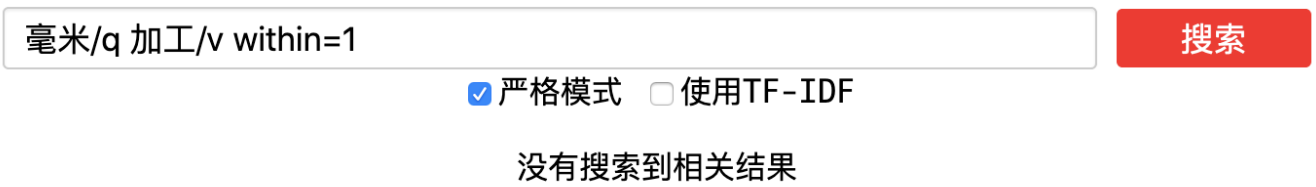
必须在严格模式打开下才生效.

within

约束词间隔



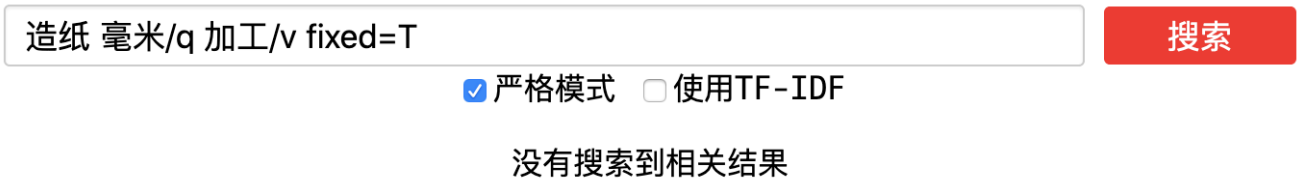
Rank	News	Score
1	0毫米），加工性能好，是造纸的首选原料。	5.7811856752628925



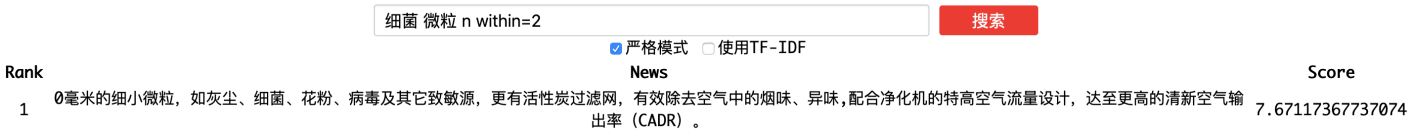
fixed

关键词在原文中的顺序必须与查询顺序一致

Rank	News	Score
1	0毫米），加工性能好，是造纸的首选原料。	9.906619614690143



词性



与上一幅图对比, 有 n, 能搜索到结果, 没有 n 就搜索不到结果, 这是因为有 n 的情况下, 允许 `细菌` `微粒` 之间再有一个名词, 比如 `灰尘`, 那么它们之间的间隔的次数可以 <=2. 没有 n, 就要求 `细菌` `微粒` 之间间隔的次数不超过 2, 但事实上不满足(间隔的词数是 4).
为了证明上面的解释. 如下图. 当 fixed=T 开启, 发现又没有结果了, 这是因为约束 n 不在中间

如果 n 在之间, 如下图, 又有结果了

实现方式

分词和词性标注使用 thulac, 索引的构建和搜索使用 Elasticsearch, 存入的结构包括 3 部分, `origin`, 表示原文, `words` 表示分词的关键词(不含词性信息), `words_poses` 含有词性信息. 实际上 `origin` 是多余的, 需要返回新闻的时候 join words 即可. 这 3 部分的类型都是 keyword, 这很重要, 否则 Elasticsearch 会自动切分, 结果完全不是想要的.

普通模式, 对 Elasticsearch 的 search 的 body 为 `bool should`, 同时设置 `minimum_should_match=1`. 严格模式 body 为 `bool must`.

关键词结合词性搜索, 是在搜索 `words_poses` 字段.

位置搜索, 用的不是 Elasticsearch. 而是得到 Elasticsearch 结果之后, 自行过滤. 实质上是, 看每一个关键词(包括词,词和词性,词性)出现的位置列表, 同一个关键词可能出现多次, 得到的是一个列表. 枚举

每一种组合, 看是否存在一种组合满足约束, 用 dfs 进行搜索.

BM25 使用的参数是, k=1.2, b=0.75. 比较看重文档的长度.

前端用 Flask 框架.

两种排序方法(TF-IDF 与 BM25)的比较与分析

TF-IDF

查询为: `"毫米/q 加工/v"`, 有的新闻里同时出现了毫米和加工两个词. 得到的排序结果为

```
390 144 292 487 3 488 499 201 212 266
```

这个结果的意思是, Elasticsearch 也会给出一个排序结果, 这个数就表示在 Elasticsearch 中的排序结果. 比如, 在 Elasticsearch 排序为 390 的在 TF-IDF 中排名第 1, 在 Elasticsearch 排序为 144 的在 TF-IDF 中排名第 2. Elasticsearch 除了前十几个之外, 剩下得分都是差不多的, 得分最高的, 是两个词都出现的, 排名 100 与排名 400 得分上也没什么差别.

结果与 Elasticsearch 本来的结果很不一样, 两个结果的前 10 仅有 1 个重合.
来分析一下什么样的情况, 在 Elasticsearch 中得分低, 但在 TF-IDF 中得分高. 390 的情况;

```
<class 'dict'>: {'origin': '0加工和冷冻食品加工。', 'words': ['0', '加工', '和',
'冷冻', '食品', '加工', '。'], 'words_poses': ['0/m', '加工/v', '和/c', '冷冻/v',
'食品/n', '加工/v', '。/w']}
```

原因是, 加工和毫米的 TF 是差不多的, 加工在这句话中出现了 2 次, 加上这句话本身很短, 因此虽然对于 `毫米` 为 0 , 但是加工却有很高的分数. 造成了这样的结果.

另一个排名较高的例子

```
<class 'dict'>: {'origin': '0对外加工。', 'words': ['0', '对外', '加工', '。'], 'w
ords_poses': ['0/m', '对外/v', '加工/v', '。/w']}
```

虽然 `加工` 只出现了一次, 但是这句话本身非常短. 因此占比很高.

但是我们的 0 号(也就是 Elasticsearch 本来的最好结果), 它输在了哪里?

```
<class 'dict'>: {'origin': '0毫米左右的鹰风筝在风筝的稳定性上，在加工煨制中，制作难易上
以及在以后的试飞中，是一种好的方案。', 'words': ['0', '毫米', '左右', '的', '鹰', '风
筝', '在', '风筝', '的', '稳定性', '上', '，', '在', '加工', '煨制', '中', '，', '制
作', '难', '易', '上', '以及', '在', '以后', '的', '试飞', '中', '，', '是', '一',
'种', '好', '的', '方案', '。'], 'words_poses': ['0/m', '毫米/q', '左右/m', '的/u'
, '鹰/n', '风筝/n', '在/p', '风筝/n', '的/u', '稳定性/n', '上/f', '，/w', '在/p',
'加工/v', '煨制/v', '中/f', '，/w', '制作/v', '难/a', '易/a', '上/f', '以及/c',
'在/p', '以后/f', '的/u', '试飞/v', '中/f', '，/w', '是/v', '一/m', '种/q', '好/a'
, '的/u', '方案/n', '。/w']}
```

很显然, 虽然它两个词都出现了, 但是, 它太长了. TF-IDF 并不看重所有关键词都出现. 虽然两个词都出现了, 如果这两个词的 IDF 差不多, 那么两个词都出现, 并不强于一个词出现 2 词. BM25 对这一点有很好的改进.
很长句子在 TF-IDF 下非常吃亏. 如果是一般的文件, 长度差异或许不会这么大, 但是 doc 都是短句子, 长句子能比短句子长上好几倍, 因此差距就很大. 我们的语料放大了长度的效果.

BM25

同样是 `"毫米/q 加工/v"` , 得到的结果如下(含义与 TF-IDF 一致)

```
3 7 2 4 6 1 390 5 144 499
```

可以看到, 仅有 3 个与 Elasticsearch 给出前 10 的结果不重合.

BM25 相对于 TF-IDF 的主要改进是 saturation, f/(f+k). 有 2 方面的效果, 当出现次数已经很高, 再增加出现次数, 得分增加的也很少, 而且增速越来越慢(score 关于 f(t,D)的导数越来越小). 这方面的效果, 对我们的情况不显著, 因为出现次数都很少, 不算饱和. 另一方面的效果是, **"出现"相比于"没出现"比"出现更多次"相比于"出现很多次"有更大的奖励**, 这一点明显不同于 TF-IDF, 也是能得到与 Elasticsearch 较一致的结果的原因.

BM25 的另一大相比 TF-IDF 的创新是文档长度如何影响分数, 引入了平均长度. 相比于平均长度长很多, 会有分数上的惩罚(通过条件 f/(f+k)中的 k). 这在这个结果中也有体现, 399 的 BM25 分值高的原因, 一方面是因为加工出现了两次, 另一方面, 它的长度短, 平均长度约为 20.89, 而它的长度为 7, 因此分数上有比较大的奖励.

144 对应的情况类似, 虽然 `加工` 只出现了 1 次. 但是长度只有 4.

针对 144 表现出来的长度问题. BM25 有参数 b 可以调节, b 越小则文档长度的奖惩越弱 .如果我们把 b 设置为 0.2, 那么理论上 144 应该排名就不高了. 结果是

```
7 2 1 3 4 6 5 0 489 499
```

489 的特点是, 长度为 61, `加工` 出现了 4 次. 由于下调了 b, 所以 489 长度大没有很大劣势, 这是为什么它之前没有出现在前 10, 现在出现了. 499 也不是长度的问题, 而是 `加工` 出现了 3 次. 因此我们看到, 如果下调 b, 长度的影响大大减小, 表现是长度小的不再靠前, 长度大的也可能靠前.

499 加工出现了 3 次, 长度为 26. 按理说, BM25 对出现次数很多的并不照顾, 当出现次数已经很多时, 再增加次数分数也不会有多大增长. 为什么同一个词出现 3 次的得分还是如此高? 这是因为我们的语料长度都很短, 关键词出现次数也都很少, 没有达到饱和, 因此 BM25 的 saturation 特性没有充分体现出来.

因此我们知道了 BM25 从希望**给的关键词都出现**的角度, 比 TF-IDF 效果好得多, 可以得出与 Elasticsearch 比较接近的结果. 但我们的语料, 长度会对结果产生很大的影响. 原因如之前所说. 长度差异可以相差几倍. 另一方面, 由于普遍长度都比较短, 次数通常最多也只出现 4 次, BM25 的 saturation 特性没充分体现.