基于不同相似度计算的排序检索算法

霍超凡

2019 年 12 月 10 日

摘要

本次实验对比了不同相似度计算方式下的检索算法，从布尔检索到向量空间模型，再到隐语义分析和主题模型，最后到基于word2vec的WMD相似度计算算法。并选择一个数据集对这些算法进行了测试，测试的结果并没有达到预期，一方面是由于自己对这些模型理解不够透彻，一些参数没有设置正确，另外还与数据集有关，这个数据集是医学方面的数据集，检索难度大，数据集给出的一些检索语句中的单词甚至没有出现在语料库中。

1 概述

检索模型经历了几次较大的变革，最初的检索模型依据检索词是否出现在文档中，这种精确匹配的方法具有内在的缺陷，空间向量模型将一篇文章看成空间中的点，根据两个向量的相似度确定文档的相似度。关于计算两个向量的相似度可以使用两个向量的距离，也可以使用夹角余弦，文章[6] 分析了这两种方法存在的问题，提出了TS-SS的向量之间相似度计算算法，克服了这两种算法劣势。LSI隐语义分析使用奇异值分解的方式，能够找出语料库中最具有代表性的词汇，将文档向量映射到这个子空间中，大幅度压缩了向量空间。文献[7] 提出了WMD计算文档相似度的方式，这种计算方法考虑到词与词之间的语义关系。另外还有LDA 主题模型，通过计算词与文档，主题与词之间的条件概率来计算文档和主题条件概率。

2 算法细节

时间关系，具体细节不再展开。

3 实验

3.1 选用的数据集

OHSUMED数据集，曾经被用于TREC-9。数据集的具体情况参考官方网址。

3.2 实验细节

根据OHSUMED所提供的数据和查询词评估模型。计算相关指标。

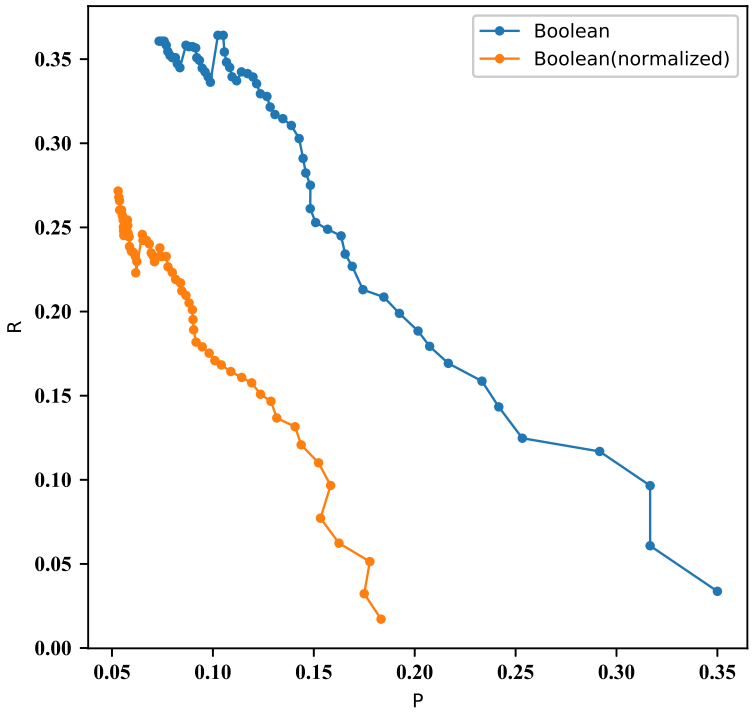


图 1: Boolean检索模型中使用夹角余弦（Boolean(normalized)）和使用向量点积（Boolean）在数据集ohsumed-87上的PR曲线。

3.3 实验结果

结果一：在向量化文档时，是否对文档的词向量进行归一化操作对结果的影响 在使用向量空间模型时，我们需要将文档表示成一个向量，向量的每一个分量是词项在文档中的权重，对于布尔检索来说，词项在文档中的权重只有0和1，词term对文档document的贡献是v_{term}

$$v_{term} = \begin{cases} 0 & term\ not\ in\ document \\ 1 & term\ in\ document \end{cases}$$

将文档中的词对文档的贡献权重依次排列组成一个向量$\mathrm{v}_{document}$，查询语句query 也按照上述规则量化为一个向量v_{query}，查询语句和文档的相似度的计算方式是夹角余弦即：

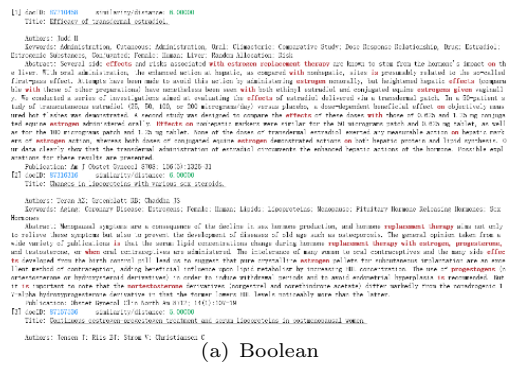
$$similarity(document, query) = \frac{\mathrm{v}_{query} \cdot \mathrm{v}_{document}}{|\mathrm{v}_{query}| \cdot |\mathrm{v}_{document}|} \tag{1}$$

另外我们也可以使用两个向量的点积作为两个向量的相似程度，即：

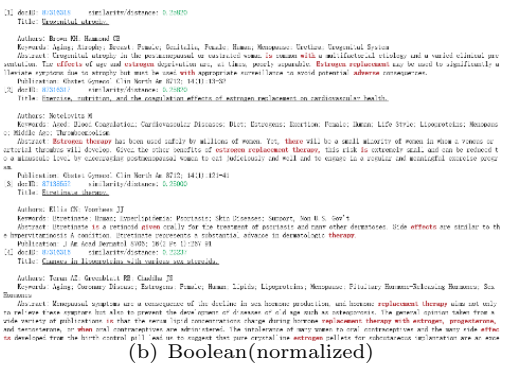
$$similarity(document, query) = \mathrm{v}_{query} \cdot \mathrm{v}_{document} \tag{2}$$

使用夹角余弦计算的相似度考虑到文档的长度，在文档与查询具有相同匹配词项数的情况下，它偏好短文章，文档的篇幅更加短小意味着它的内容更专一，可是在OHSUMED的查询中，被归一化后的计算方式，性能反而下降。如图7所示，经过归一化操作后的检索结果明显不如进行向量点积计算的结果。通过观察结果，我们发现在数据集ohsumed-87 中篇幅短小的并不是最具有相关性的，例如图8，篇幅短小的文档被排在第一位，可是和查询相关的被排在了第三位，篇幅短小的优势冲淡了文档与查询词之间的匹配程度。

这种现象仅仅出现在布尔检索模型中，对于其它例如使用词频作为权重和使用逆文档词频作为权重进行量化文档的方法，归一化后，效果明显提升。例如图3所示，无论是使用



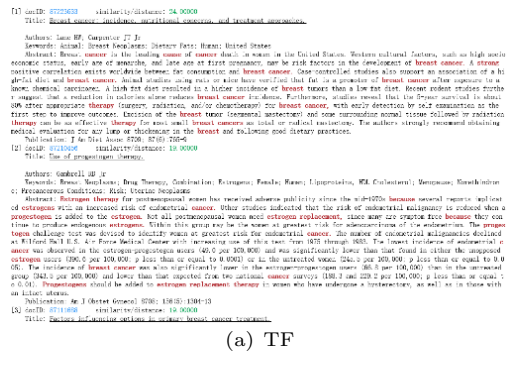
(a) Boolean



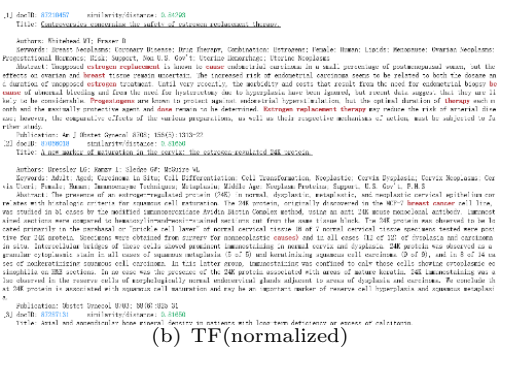
(b) Boolean(normalized)

图 2: 使用数据集OHSUMED-87上的OHSU1查询语句对两种不同相似度计算方式的布尔模型测试，图2(a)是使用向量点积进行相似度计算得到的查询结果，图2(b)是使用夹角余弦计算相似度得到的结果，输入的查询词为“Are there adverse effects on lipids when progesterone is given with estrogen replacement therapy”，左图中第二篇文档与查询词相关，而右图中第四篇和查询词相关，右图中排在前面的都是篇幅短小的文章，受篇幅的影响，即使包含查询语句较少的单词，处于篇幅短小的优势仍然可以得到很高的分数。

词频TF计算权重还是使用逆文档词频TF-IDF计算权重都能提高检索的性能。同样的，分析查询结果发现，文档中频繁出现查询词的个别词并不是最相关的。



(a) TF



(b) TF(normalized)

图 3: 使用数据集OHSUMED-87中的测试集提供的查询语句ohsu-5：does estrogen replacement therapy cause breast cancer，测试TF检索模型，左图是没有经过归一化操作的检索结果，右图经过归一化。由于没有经过归一化操作，频繁出现查询词占有较大的优势，左图的排在第一位的文档中频繁出现breast cancer，但是这篇文章和estrogen没有关系，而右图中仅仅出现依次breast cancer却和查询词相关。

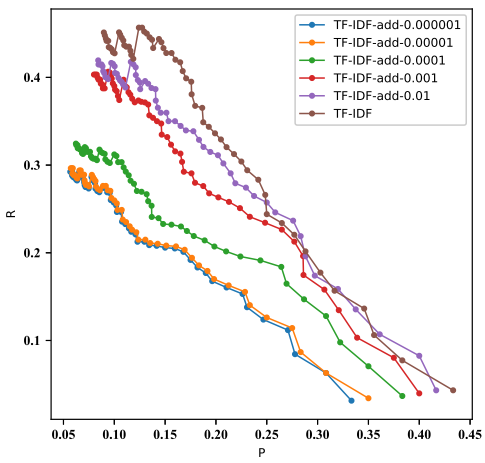
结果二：IDF相比TF更加能够反映查询词和文档的相关程度　我使用不同的方式结合TF和IDF，并且通过调整参数使IDF和TF对词语的权重的影响比例，我使用了两种方式结合TF和IDF，一种方式是将两者相加，

$$w_{term} = \alpha \text{IDF} + (1-\alpha)\text{TF} \tag{3}$$

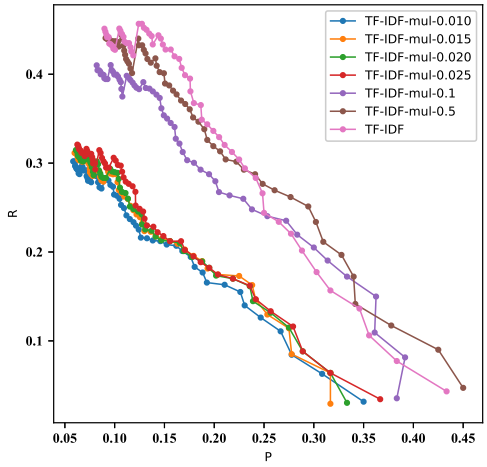
式3中通过逐渐增大α，可以加大IDF对词权的影响。另外一种方式是使用乘法，

$$w_{term} = \text{IDF}^{\alpha} + \text{TF}^{1-\alpha} \tag{4}$$

通过增大α仍然可以增大IDF的比重。



(a) TF-IDF-mul



(b) TF-IDF-add

图 4: 不同词语权重的计算方式对TF-IDF检索模型的影响，测试的数据集仍然是OHSUMED-87，所选用的检索语句来自于测试集中的ohsu类。图中图例的标注后面的数字对应于式34中的α，没有参数标注的默认使用log计算。

```
[1] docID: 87310088     similarity/distance: 7.13145
    Title: Bilateral carcinoma of the middle ear.

    Authors: Milford CA; Violaris N
    Keywords: Aged; Carcinoma, Squamous Cell; Case Report; Ear Neoplasms; Ear, Middle; Human; Male
    Abstract: We describe the first reported case of bilateral carcinoma of the middle ear.
    Publication: J Laryngol Otol 8712; 101(7):711-3
[2] docID: 87183749     similarity/distance: 8.48447
    Title: Patch repair of the left ventricular free wall following aneurysmectomy.

    Authors: McGiffin DC; Kirklin JK
    Keywords: Adult; Aged; Heart Aneurysm; Heart Ventricle; Human; Implants, Artificial; Male; Middle Age; Polyethylene Terephthalate; Posto
perative Period; Stroke Volume
    Abstract: A technique of reconstruction of a left ventricular free wall defect, which has been used in 7 patients, is described. Situati
ons in which its use may be appropriate are discussed.
    Publication: Ann Thorac Surg 8707; 43(4):441-2
[3] docID: 87179501     similarity/distance: 8.73147
    Title: The unbreakable stone?

    Authors: Plawner J; Surya BV; Rothberg M
    Keywords: Case Report; Human; Kidney Calculi; Kidney Pelvis; Lithotripsy; Magnesium; Male; Middle Age; Phosphates; Time Factors
    Abstract: A case of a failed percutaneous ultrasonic lithotripsy is reported as an unusual occurrence.
    Publication: Urology 8707; 29(4):400-1
```

图 5: 仍然使用查询词：“does estrogen replacement therapy cause breast cancer”，但是使用TS-SS计算得到的相似度较高的文档均是短文档，TS-SS似乎是更看重文章篇幅。

结果三：TS-SS不适合用于计算短查询词和长篇文章的相似程度 TS-SS的设计理念就在于克服了夹角余弦和欧式距离的各个劣势，使得两个向量不仅仅方向相同，还要具有近乎相同的模长。只有两个向量完全重叠时TS-SS的值才为0，在实验中盲目使用新方法，结果性能极差，没有考虑到查询词和文档内在的篇幅差异。

结果四 本次实验中使用的LSI模型，即使将维度升到3000，仍然性能很差，不如Boolean模型，如图6所示，提高维度的确会提高性能，但是程度并不显著。

结果五：传统的使用TF-IDF仍然具有无与伦比的优势 在各个数据集上TF-IDF 都明显优于其它方法，而一些现代方法如LSI、LDA，可能是我实验观察中对这些算法内在机理的理解并不到位，并没有达到需要的结果。

结果六 使用WMD计算文档相似度运算量极大，适合离线处理文档的分类问题，无法满足实时查询的要求，而且WMD计算的是从一篇文档通过语义变化替换一些词，变为另一

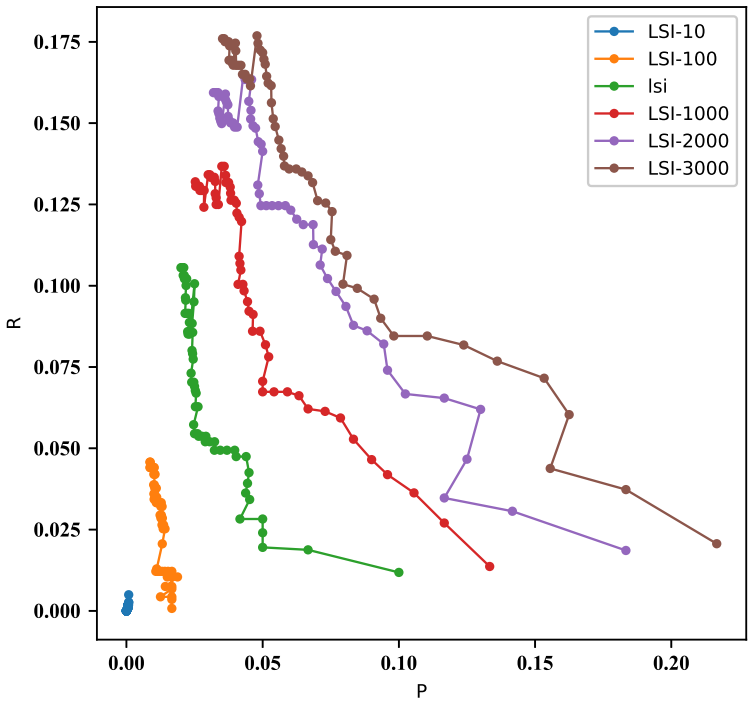
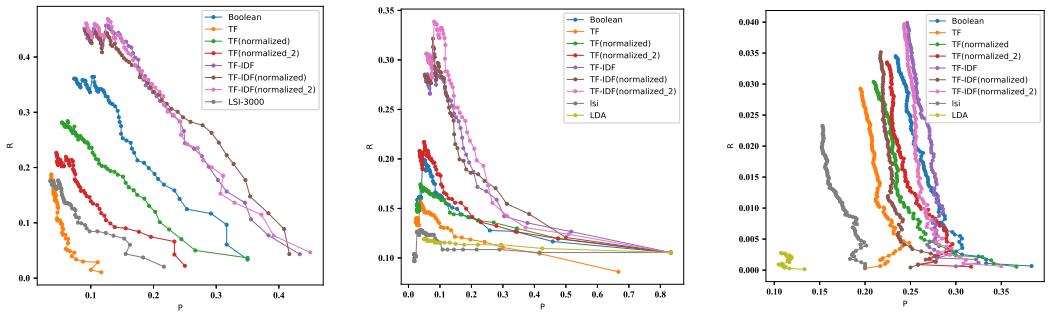
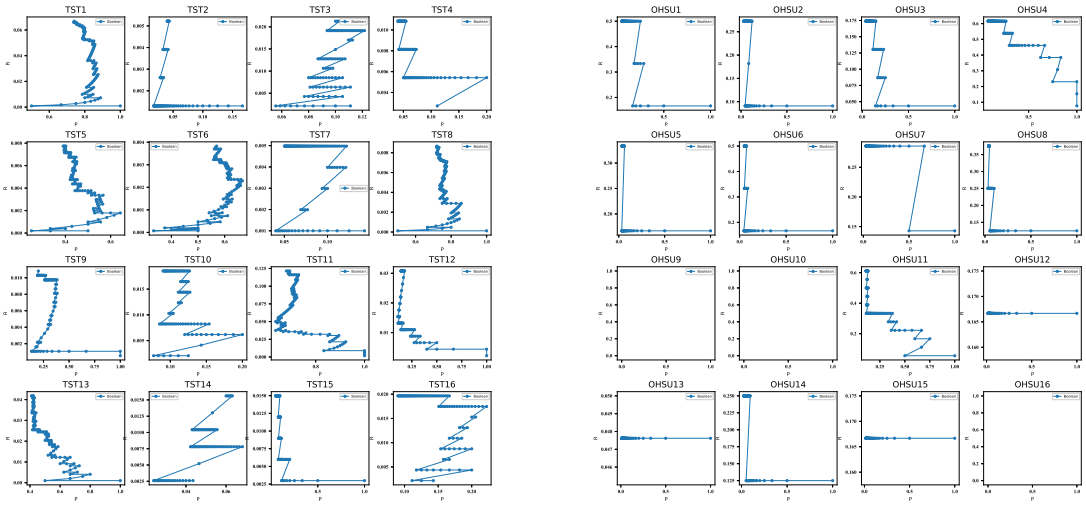


图 6: LSI模型在数据集OHSUMED-87中的性能随着模型向量化文档的维度的关系图



(a) OHSU-PR (b) OHSU-PR-2 (c) OHSU-mesh-test-PR

图 7: 不同检索模型在不同数据集下的PR曲线，以上三幅图均是在OHSUMED-87上测试的结果，图7(a)使用测试集所提供的ohsu类查询语句进行查询，图 7(b) 使用一篇完整的文档进行查询，所以刚开始精度很高，之后迅速下降，图7(c)使用测试集提供的mesh 类的查询词进行查询。



(a) Boolean-PR-mesh (b) Boolean-PR-ohsu

图 8: Boolean检索模型在不同查询语句的PR曲线，左图是在OHSUMED-88-91上的mesh查询语句的测试结果，右图是在OHSUMED-87的训练集ohsu查询语句集测试的结果。可以发现，Boolean检索模型对检索词十分敏感，有时甚至降为0。

篇文档的代价，不适于短查询和长篇文章之间的匹配。

结果七 不同模型对查询词十分敏感，像Boolean、TF-IDF不能适用于处理一些复杂的查询语句。有时Boolean检索模型优于复杂的检索模型。

表 1: 各种模型在数据集OHSUMED-87的训练集查找词ohsu上的各个检测指标。

	P@10	R@10	F1@10	CG@10	DCG@10	nDCG@10	MAP@10
Boolean	0.201	0.188	0.181	3.316	2.748	0.219	0.262
TF(normalized)	0.168	0.158	0.153	2.783	2.184	0.172	0.221
TF-IDF	**0.249**	**0.244**	**0.231**	**4.000**	**3.225**	**0.258**	**0.321**
LSI-3000	0.093	0.089	0.085	1.433	1.268	0.102	0.143v

表 2: 各种模型在数据集OHSUMED-87的测试集mesh所提供的检索词上的各个检测指标。

	P@10	R@10	F1@10	CG@10	DCG@10	nDCG@10	MAP@10
Boolean	**0.305**	**0.005**	**0.009**	**3.066**	**2.203**	**0.200**	**0.320**
TF(normalized)	0.278	0.004	0.008	2.783	2.088	0.189	0.331
TF-IDF	0.274	0.004	**0.009**	2.75	2.039	0.185	0.289
LSI-500	0.189	0.003	0.006	1.9	1.366	0.124	0.191

4 文献评注

[7]是关于WMD的文章，原文讲WMD用于文本分类中，我将其应用到检索系统发现结果没有预期的那样好。[6]是关于TS-SS的文章，这里面分析了欧式距离的夹角余弦在测量

文档距离时的缺陷。[1]实现了检索模型的核心算法，我将其做了较大的改动，一些地方保留了原作者的编程思路。[3]中实现了WMD计算两个句子的相似度。[2][4][5]分别是实现TS-SS、LDA、LSI 的代码，我在实验中较大篇幅的参考了这几篇代码。

参考文献

[1] advanced-ir-search-engine. `https://github.com/mdietrichstein/advanced-ir-search_engine`. Accessed December 09,2019.

[2] Vector similarity. `https://github.com/taki0112/Vector_Similarity`. Accessed December 09,2019.

[3] Word mover distance. `https://github.com/PragmaticLab/Word_Mover_Distance`. Accessed December 09,2019.

[4] Document-similarity. `https://github.com/khoaipx/Document-Similarity/tree/master/Document-Similarity`. Accessed December 09,2019.

[5] python-lsi-similarity. `https://github.com/neomoha/python-lsi-similarity`. Accessed December 09,2019.

[6] Arash Heidarian and Michael Dinneen. A hybrid geometric approach for measuring similarity level among documents and document clustering. pages 142–151, 03 2016.

[7] Matt Kusner, Y. Sun, N.I. Kolkin, and Kilian Weinberger. From word embeddings to document distances. *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, pages 957–966, 01 2015.