一种改进的 Lucene 语义相似度检索算法*

黄承慧[1,2]，印 鉴[1]，陆寄远[2]
（1. 中山大学信息科学与技术学院，广东 广州 510275；
2. 广东金融学院计算机科学与技术系，广东 广州 510520）

摘 要：在 Lucene 的基础上，结合检索词项的语义信息，利用外部词典 Wordnet 分析检索词项与被检索文档中词项的语义相似度，在此基础上实现对文档语义信息的检索。通过分析现有的相似度量函数的核心特征，选择合适的语义相似度量方法，提出了一种新的词项语义相似度检索函数，该函数能够对检索文档按照语义相似度进行排序。实验结果表明，所提出的方法能够有效地提升文献检索的准确度。

关键词：语义；相似度；信息检索；算法
中图分类号：TP311 **文献标志码**：A **文章编号**：0529-6579（2011）02-0011-05

An Improved Retrieve Algorithm Incorporated Semantic Similarity for Lucene

HUANG Chenghui[1,2], *YIN Jian*[1], *LU Jiyuan*[2]
（1. School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275, China;
2. Department of Computer Science and Technology, Guangdong University of Finance, Guangzhou 510520, China）

Abstract: A retrieve algorithm that incorporates the semantic information of the words into traditional retrieve function of Lucene is proposed. The proposed method improves the important components of existing retrieve similarity functions with semantic information, and selects the appropriate measure of semantic similarity to compute the semantic similarity between the query words and text corpus by using the external dictionary Wordnet. With the semantic similarity, the algorithm implements semantic information retrieve and can sort the retrieved text documents according to the semantic similarity between query words and text documents. The experimental results show that the proposed method can improve the precision of document retrieval effectively.

Key words: semantic; similarity; information retrieve; algorithm

随着信息时代的到来，几乎所有的纸质文件都将转化为电子版进行保存。这种趋势是不可逆转的，对比纸质文件，电子文件更容易保存和使用，同时也更安全。所有这些要求促使我们寻找适当的方法，帮助用户在日益广阔的信息海洋中更加有效的浏览和管理这些电子文件。

Lucene 是一个基于 Java 的用于文本检索和搜索的开放源代码工具包[1]。应当指出的是，Lucene 不是一个具备完整功能的搜索应用程序，它只是一个具备索引和检索功能的软件库，相对于其他检索

* **收稿日期**：2010-03-31
基金项目：国家自然科学基金资助项目（60573097，60773198，60703111）；广东省自然科学基金资助项目（05200302，06104916）；广州市科技计划资助项目（2007Z3-D3071）；高等学校博士学科点专项科研基金资助项目（20050558017）；新世纪优秀人才支持计划资助项目（NCET-06-0727）
作者简介：黄承慧（1976年生），男，博士生，讲师；E-mail：hch. gduf@163. com

工具，Lucene 有着更好的灵活性，人们可以将其应用于文献检索和搜索引擎等。

尽管 Lucene 已经被广泛应用，相关的研究也层出不穷，然而大多数研究都是基于 Lucene 内部默认实现的词频分析检索函数来考察检索文本之间的相似性以进行检索，很少有考虑词项语义的 Lucene 检索研究。此外，对于 Lucene 词频分析检索函数的性能也很少被讨论。因此，如果能对 Lucene 的检索函数加以改进的话，则能够有利于各种基于 Lucene 的应用，如业界广泛使用的开源搜索引擎 Nutch[2]。

本文基于上述观察，提出了一种结合检索词项语义的检索函数，该函数改进了传统基于词频的方法对语义忽视所造成的检索不够精确的问题，同时也给出了一个初步判定文档相似性的算法。通过这些改进，实验结果表明，对比传统的基于词频的方法，本文提出的方法能够取得较好的检索精确度和召回率。

1 相关工作

Lucene 作为优秀的检索软件包，被广泛的应用到检索领域中[3-5]。但其用于处理检索结果排序的核心技术是基于传统词频分析技术的，因而在检索精确度和召回率上存在先天的不足。针对这些问题，许多文献都利用附加的外部本体来改进 Lucene 的检索工作。

文献［6］为了解决传统搜索引擎所面临的低效检索精确度以及无法理解用户查询意图的缺陷，在 Lucene 的基础上提出了一个基于本体的语义搜索引擎框架。文献［7］在开放目录项目（Open Directory Project）提供的轻量级本体的基础上，利用经典的 TF－IDF 词频分析技术，Lucene 用于计算检索结果和相关概念的相似度，以此提高搜索结果的质量。文献［8］则根据 Wiki 百科辞典的问答任务，分析 Wiki 百科辞典中的文章和查询中的单词词频，利用 Lucene 计算相似度，获得了较好的结果。

在各种本体的使用上，Wordnet 是使用最多的一个本体知识库。文献［9－10］都是利用 Wordnet 提供的同义词对用户的查询进行词项扩展，同时优化用户提交查询语句中的关键词组，以此提高 Lucene 的检索效果。文献［11］利用潜语义索引技术和 Wordnet 作为本体，提出了一种独立于 Wordnet 知识但依赖于潜语义索引知识的模型，利用该模型扩展 Lucene 查询，取得了较好的效果。类似的，文献［12］则集成了 Lucene、WORDNET、潜语义分析以及和领域相关的受控词汇等技术来改进 Lucene 检索结果的有效性。

面对冗长的搜索结果列表，用户通常只会选择排在前面若干项的搜索结果而忽略后面的搜索列表。文献［6］提出了一种高效的搜索方法来减少搜索结果的冗长列表。通过文中提出的基于本体的语义自适应搜索技术，本体中储存了检索词项之间的关系，在用户查询被 Lucene 处理之前，该方法首先将用户的查询扩展为本体中的词汇及其关系，并对扩展后的词汇进行加权处理，每次用户对检索结果的点击浏览都回引起权重的改变，从而逐步逼近用户真实的查询意图，减少了不必要的检索结果。

上述种种对 Lucene 检索的扩展研究，或者考察利用本体对检索词项的词频进行分析，或者利用本体考察检索词项在本体中所处层次结构的相似性，均未考察检索词项本身的语义相似性。因而不能更好的理解用户检索所涉及的真正意图。基于此，本文在 Lucene 内置的词频相似度函数的基础上考察词项的语义相似性，以此提高检索的准确度和召回率。

2 Lucene 相似度函数的改进

Lucene 内部缺省实现的相似度检索函数来源于经典的词频分析技术 TF－IDF。该方法将文本看作是一个容纳词项的袋子，不考虑词项出现的顺序，也不考虑词项的含义。文本特征向量由文本中出现的词项在单个文本中出现的频率以及该词项在整个文本集中出现的频率来表示。每一篇文本建模为文中出现的 n 个加权词项组成的向量。该方法基于以下经验观察

1）词频（Term Frequency）：某个词项在一个文本中出现的次数越多，它和文本的主题越相关；要注意在特定的语言环境下都有许多特定的词不具备这种特性而应将其排除，如中文的“的”“地”、英文的“a”“an”。

2）逆文本频率（Inverse Document Frequency）：某个词项在文本集合的多篇文本中出现越多，该词项的区分能力越差。例如：在一个包含 1000 篇文本的集合中，如果某个词项 A 在 100 篇文本中都出现，而另一个词项 B 只在 10 篇文本中出现，则词项 B 比 A 具有更好的区分能力。

通过对文本集合中的每一个词项都进行上述分析，得到每一篇文本中每一个词项的 TF－IDF 值。

之后再利用这些 TF - IDF 值为每一篇文本建立一个向量模型，通过计算向量间的余弦相似度或者 Jaccard 系数来表示文本之间的相似性，最终根据检索文档与用户查询之间的相似度值高低排序，将检索结果以列表形式返回给用户。

Lucene 的评分机制采用了信息检索中的空间向量模型和布尔模型相结合的方法，来最终决定一个给定的文档和一个用户的查询到底从多大程度上是相关的。在查询中，使用布尔模型中的布尔逻辑首先减少了需要进行评分的文档。其次，Lucene 在支持布尔模型和模糊查询的基础上，还加入了一些性能提高和优化。但是其核心还是基于向量模型。

Lucene 采用的相似度计算函数如公式（1）所示

$$S(Q,D) = \frac{|Q \cap D|}{|Q|} \times \mathrm{qNorm}(Q) \times \sum_{t \in Q} \left(c(t,D)^{\frac{1}{2}} \times 1 + \log \frac{N}{\mathrm{df}(t) + 1}\right)^2 \times |D| \times \mathrm{Boost}(t,D)) \tag{1}$$

上式中 Q 代表用户查询，D 代表被检索的文档，两者均被表示为词频向量。$c(t, D)$ 表示词项 t 在文档 D 中出现的频率。$\mathrm{df}(t)$ 是文档集合中包含词项 t 的文档数目，$|D|$ 是文档 D 的长度，$|Q|$是查询 Q 的长度，$|Q \cap D|$是同时出现在文档 D 和查询 Q 中的词项的数目，Boost (t, D) 表示与查询词项 t 和查询 D 有关的加权因子。qNorm (Q) 是一个规范化因子，在搜索的时候起作用，使得不同查询间的分数可比较。其计算公式如下

$$\mathrm{qNorm}(Q) = \sum_{t \in Q} \left(\mathrm{qBoost}(t,Q) \times \left(1 + \log \frac{N}{\mathrm{df}(t) + 1}\right)^2\right) \tag{2}$$

其中 qBoost (t, Q) 表示与查询词项 t 和查询 Q 有关的加权因子。

文献［13］对信息检索的相似度函数做了深入研究，提出了一种更健壮、更好的基于公理化思想的检索相似度函数。其计算公式如下

$$S(Q,D) = \sum_{t \in Q} c(t,Q) \cdot \left(\frac{N}{\mathrm{df}(t)}\right)^{0.35} \cdot \frac{c(t,D)}{c(t,D) + 0.5 + \frac{0.5 \cdot |D|}{\mathrm{avdl}}} \tag{3}$$

其中 $c(t, Q)$ 表示词项 t 在查询 Q 中出现的频率，$\mathrm{df}(t)$ 是文档集合中包含词项 t 的文档数目，$c(t, D)$ 表示词项 t 在文档 D 中出现的频率，$|D|$ 表示文档 D 中所有不同词项的总数目，即文档词项向量的维度，avdl 则表示整个文档集合中所有文档包含的平均词项数目。

尽管上述检索函数在实践中证明有着较好的效果，然而他们都未能捕捉文本中的语义信息。随着互联网的发展，海量的文本数据要求我们能够更为精确的捕捉和刻画文本的含义而不仅仅是文本词项出现的频率。例如一篇关于银行（bank）的文章和一篇关于河岸（bank）的文章，由于银行和河岸两者词项的拼写都是 bank，基于词频的检索方法就会将它们检索返回给用户。而一篇关于苹果和一篇关于橘子的文章则因为两者的词项拼写不同（apple 和 orange）在用户检索时只返回苹果或者橘子的文章，而无法同时返回给用户。

如果考察检索词项的语义信息，更为准确的捕捉用户的意图就能够有效地解决上述不足。为此，本文针对公式（3）结合词项语义相似度进行改进，改进后的检索函数如公式（4）所示

$$S(Q,D) = \sum_{t \in Q} \mathrm{Sim}(t,Q) \times \left(\frac{N}{\mathrm{Sim\_df}(t)}\right)^{0.35} \times \frac{\mathrm{Sim}(t,D)}{\mathrm{Sim}(t,D) + 0.5 + \frac{0.5 \cdot |D|}{\mathrm{avdl}}} \tag{4}$$

其中 $\mathrm{Sim}(t, Q)$ 表示查询 Q 中与词项 t 相似的词的频率，$\mathrm{Sim\_df}(t)$ 表示文档集合中包含与词项 t 相似的文档数目，$\mathrm{Sim}(t, D)$ 表示文档 D 中与词项 t 相似的词项的频率。这三个涉及词项相似度的因子要求我们给出词项相似度的阈值，我们将在实验部分对其进行阐述。

与公式（3）比较，本文扩展了经典的词频分析技术，使得检索结果不仅包含了被检索词项出现的文档，而且还包含了与被检索词项相似的文档，从而更为准确的体现了检索的含义。我们注意到，由于相似性的引入，使得原本只由检索词项出现的检索结果扩大为与检索词项具有相似性的检索结果，从而大大的增加了检索返回的文档数目。因此，我们有必要对返回的检索结果进行适当的过滤处理，使得检索结果既包含与用户检索需求相关的文档，又不至于泛滥到包含所有与用户检索相关甚小的文档。具体的算法描述如下

算法 1：

1）预处理文档集合，删除停用词。

2）初始化文档向量。

3）遍历文档集合，不考虑相似度计算 $\mathrm{df}(t)$ 。

4）遍历文档集合根据词项相似度计算方法寻找包含与检索词项相似度超过阈值 μ（$0 < \mu \leqslant 1$）

的词汇的文档。

5）根据公式（4）计算文档与用户查询的相关性，按相关性大小降序返回检索列表 PrimaryList。

6）取检索列表 PrimaryList 排名在 $df(t)/\mu$ 之前的文档作为最终检索结果 ResultList。

算法 1 中用于计算词项之间相似度的方法，采用了 WordNet::Similarity 工具包，该工具包实现了 8 种主流的词与词之间相似度计算的方法[14]。文献［15］指出，基于信息内容度量的相似度方法优于其它方法。因此，本文采用了文献［16］所实现的相似度算法作为算法 1 中计算词项之间相似度的方法。相似度阈值 μ 在实验部分有详细描述。

3 实 验

实验数据采用业界广泛采用的 Ruters－21578 数据集（Re1），Re1 数据集采用了 ModApte 划分，共 9 603 篇文档，Ruters－22173 数据集（Re2），Re^2 数据集采用了 ModWiener 划分，共 9 610 篇文档。在数据集 Re1，Re2 上建立 Lucene 支持的布尔查询，对 Lucene 缺省的相似度函数、改进的公理化相似度函数以及本文提出的基于语义相似性的度量函数进行了对比研究，实验结果如图 1 所示。

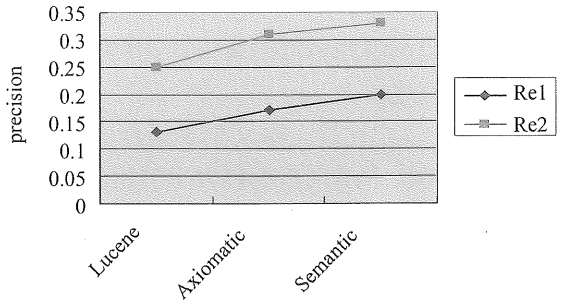

图 1 三种相似度函数的对比实验结果

Fig. 1 Effect comparison of three similarity function

本文提出的语义相似度检索公式（4）中，需要根据词项的相似度来计算查询 Q 中与词项 t 相似的词的频率、文档集合中包含与词项 t 相似的文档数目以及文档 D 中与词项 t 相似的词项的频率。实验采用了基于信息内容度量的相似度计算方法，在对文档集合的常见词汇进行计算后，我们设定词项相似度阈值为 0.57，即两个词项之间的相似度如果超过 0.57，则认为这两个词项是相似的，从而对相应的词频进行计数。需要指出的是，计算两个词项相似度的 WordNet::Similarity 工具包是采用 Perl 语言编写的，计算效率较低。因此，本文对实验数据集中的词项相似度预先进行了计算，以名称为两个词项，值为词项相似度的哈希表的形式保存在内存中，实际进行检索时，只需从哈希表中查找已经计算好的相似度，从而减少算法的运算时间。

实验的总体结果是比较满意的，在 Re1，Re2 两个数据集上，对比传统的基于词频的相似度函数，本文提出的方法在检索精度指标上均取得了 10% 以上的提升。这说明了本文在传统词频分析的基础上结合语义信息是可行的。

4 结 语

本文针对 Lucene 内置的文本检索相似度函数进行了改进，将传统的基于词频分析的相似度函数增加词项的语义信息，并利用基于信息理论的词项语义相似度量理论计算词项之间的语义相似性，以此对文档进行检索。实验结果表明，本文提出的方法是有效的，能够进一步提高检索的精度。

本文对文本检索的语义信息进行了有益的尝试，实验的结果虽然对比传统方法有一定的改进，但计算词项之间相似度的时间开销是需要我们进一步去优化和改进的。此外，单纯的考察词项之间的相似性丢失了文档内在的结构性特征，以致检索结果的提高有一定的局限性。我们预期在后续的研究里进一步考察文档的结构信息，并结合现有的语义相似度分析技术，以便得到更好的检索效果。

参考文献：

[1] Lucene. Lucene Java 3.0.1[EB/OL].(2010－02－26)[2010－03－30]. http://lucene.apache.org/

[2] Nutch. Apache Nutch 1.0 release[EB/OL].(2009－03－23)[2010－03－30]. http://lucene.apache.org/nutch/

[3] SONG J, ZHU Y Q, LIU R D. Enhanced full text retrieval kit based on Lucene [J]. Computer Engineering and Applications, 2008, 44 (4): 172－175.

[4] GUAN J H, GAN J F. Design and implementation of web search engine based on Lucene [J]. Computer Engineering and Design, 2007, 28(2):489－491.

[5] ZHOU D P, XIE K L. Lucene search engine [J]. Computer Engineering, 2007, 33(18):95－96.

[6] YANG C, YANG K C, YUAN H C. Improving the search process through ontology-based adaptive semantic

search [J]. Metadata and Semantics for Digital Libraries, 2007, 25(2) : 234 - 248.

[7] ZHU D Y, DREHER H. Determining and satisfying search users real needs via socially constructed search concept classification [C]//IEEE DEST 2007, 2007.

[8] BUSCALDI D, ROSSO P. A bag-of-words based ranking method for the Wikipedia question answering task [C]// CLEF 2006, 2007: 550 - 553.

[9] ZHENG T, ZHENG C. Semantic retrieval system based on Lucene [J]. Computer Engineering, 2008, 34(16): 92 - 94.

[10] JIANG Y F, WANG H, ZHANG Y H, et al. Design and implementation of semantic search engine based on Lucene[J]. Computer Engineering and Design, 2008, 29(20):5336 - 5341.

[11] DU L, JIN H D, DE VEL O, et al. A latent semantic indexing and WordNet based information retrieval model for digital forensics[C]// IEEE ISI 2008, 2008: 70 - 75.

[12] RAVISHANKAR D, THIRUNARAYAN K, IMMANENI T. A modular approach to document indexing and semantic search[C]// WTAS 2005, 2005: 165 - 170.

[13] FANG H, ZHAI C. An exploration of axiomatic approaches to information retrieval [C]//Proceedings of the 2005 ACM SIGIR Conference on Research and Development in Information Retrieval, 2005: 480 - 487.

[14] PEDERSEN T, PATWARDHAN S, MICHELIZZI J. Wordnet:: similarity-measuring the relatedness of concepts[C]//Proc of AAAI - 04, San Jose, California, USA, 2004: 1024 - 1025.

[15] BUDANITSKY A, HIRST G. Semantic distance in WordNet: an experimental, application-oriented evaluation of five measures [C]// Proc of the Workshop on WordNet and other Lexical Resources, 2001.

[16] LIN D. An information-theoretic definition of similarity [C]//Proceedings of the 15th International Conference on Machine Learning, Madison, WI, US, 1998.

(上接第 10 页)

参考文献：

[1] HARALICK R M, SHANMUGAN K, DINSTEIN I. Texture features for image classification [J]. IEEE Trans Syst, Man Cybern, 1973, 3(6): 610 - 621.

[2] PORTER R, CANAGARAJAH N. Robust rotation-invariant texture classification: wavelet, Gabor filter and GMRF based schemes [J]. IEE Proceedings in Vision, Image and Signal Processing, 1997, 144(3): 180 - 188.

[3] CHELLAPPA R, CHATTERJEE S. Classification of textures using Gaussian Markov random fields [J]. IEEE Trans Acoust, Speech, Signal Processing, 1985, 33(4): 959 - 963.

[4] LIU X, WANG D. Texture classification using spectral histograms [J]. IEEE Trans Image Processing, 2003, 12(6): 661 - 670.

[5] LAURITZEN S. Graphical models [M]. Oxford Statistical Science Series. Oxford: Oxford University Press, 1996.

[6] JORDAN M I. Graphical models[J]. Statistical Science, Special Issue on Bayesian Statistics, 2004, 19: 140 - 155.

[7] RUE H, HELD L. Gaussian Markov random fields: theory and applications [M]. Chapman & Hall/CRC, 2005.

[8] TIBSHIRANI R. Regression shrinkage and selection via the lasso [J]. J Roy Stat Soc, B, 1996, 58: 267 - 288.

[9] ZOU H, HASTIET. Regularization and variable selection via the elastic net [J]. J Roy Stat Soc, B, 2005, 67: 301 - 320.

[10] ZOU H. The adaptive Lasso and its oracle properties [J]. J American Stat Assoc, 2006, 101(476):1418 - 1429.

[11] HOERAL A, KENNARD R. Ridge regression [M]. In Encyclopedia of Statistical Sciences. New York: Wiley, 1988:129 - 136.

[12] EFRON B, HASTIE T, JOHNSTONE I, et al. Least angle regression [J]. Ann Stat, 2004, 32(2): 407 - 451.

[13] WANG H, LI B L, LENG C. Shrinkage tuning parameter selection with a diverging number of parameters [J]. J Roy Stat Soc, B, 2009, 71(3): 671 - 683.

[14] WEBER A G. The USC-SIPI image database [M]. Version 5. USC-SIPI Report, Signal and Image Processing Institute, 2006.