计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190800139

面向加密云数据的多关键字语义搜索方法

李 彦 申德荣 聂铁铮 寇 月
东北大学计算机科学与工程学院 沈阳 110169
(nanou1995@163.com)

摘 要 由于云服务具有灵活性、通用性和低成本等特性,将数据交由云服务器管理变得日益普遍。然而,云服务器不是完全可信的,因此将加密数据交由云服务器管理并支持加密搜索成为了当前研究的热点问题之一。加密虽然能够很好地保护数据隐私安全,但是会掩盖数据本身的语义信息,加大搜索难度。文中面向加密云数据提出了一种支持多关键字的安全语义搜索解决方案,其核心思想是基于主题模型获取文档的主题向量和主题的词分布向量,通过计算查询关键字与各个主题的语义相似度生成查询向量,支持在同一向量空间内评价查询向量与文档主题向量的相似度;提出了基于EMD并结合词嵌入计算查询向量与主题相似度的方法,提升了查询关键词与主题之间语义相似度的准确性;为支持高效语义搜索,构建了主题向量索引树,并采用"贪婪搜索"算法优化关键字搜索。理论分析和实验结果表明:所提解决方案可实现安全的多关键字语义排序搜索,并且大大提高了搜索效率。

关键词:加密可搜索;语义搜索;隐私保护;云计算;查询处理
中图法分类号 TP391

Multi-keyword Semantic Search Scheme for Encrypted Cloud Data

LI Yan,SHEN De-rong,NIE Tie-zheng and KOU Yue
College of Computer Science and Engineering,Northeastern University,Shenyang 110169,China

Abstract Due to the flexibility,versatility,and low cost of cloud services,it is common to hand over data to cloud server management. However,cloud servers are not completely trusted,so it is one of the hot issues in current research to transfer encrypted data to cloud servers and support encrypted search. Although encryption can protect data privacy and security,it will cover the semantic information of the data itself and increase the difficulty of searching. This paper proposes a secure semantic search solution for multi-keywords for encrypted cloud data. The core idea is to obtain the topic vector of the document and the word distribution vector of the topic based on the topic model,and calculate the query keyword to be similar to the semantics of each topic. The query vector is generated to support the similarity between the query vector and the document subject vector in the same vector space. The calculation method of calculating the similarity between the query vector and the topic based on EMD combined with word embedding is proposed to improve the accuracy of semantic similarity. To support efficient semantic search,a topic vector index tree is constructed and a "greedy search" algorithm is used to optimize keyword search. Finally,theoretical analysis and experimental results show that the proposed solution can achieve secure multi-keyword semantic sorting search and greatly improve search efficiency.

Keywords Encryption searchable,Semantic search,Privacy protection,Cloud computing,Query processing

1 引言

云服务具有的灵活性、通用性和低成本等特性,使得越来越多的公司和组织愿意将数据外包给云服务商管理,其中可能包含敏感信息,如私人健康记录、电子邮件、公司财务报表等。然而,云服务器不是完全可信的,可能造成用户隐私数据的泄露。为了保护隐私,用户往往将数据加密后再外包给云服务器。因此,如何在加密数据中检索符合用户需求的数据成为了备受关注的问题。

针对上述问题,学术界已经提出了一些解决方法,如采用完全同态加密[1]和可信硬件[2]的方法,但是这些方法的复杂度或者高成本限制了其在云环境中的应用。还有一些在云环

到稿日期:2019-08-28 返修日期:2019-11-29 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61672142,U1811261);国家重点研发项目(2018YFB1003404);中央高校基本科研业务费(N171606005)
This work was supported by the National Natural Science Foundation of China(61672142,U1811261),National Key R&D Program of China(2018YFB1003404) and Fundamental Research Funds for the Central Universities (N171606005).
通信作者:申德荣(shendr@mail.neu.edu.cn)

境的多种威胁模型下实现不同搜索功能的成果，包括多关键字的排序搜索和模糊查询等，但是现有工作只考虑了关键字字符的模糊搜索，忽略了关键字之间的语义关系。

本文基于 LDA(Latent Dirichlet Allocation)主题模型，提出了一种高效安全的语义搜索解决方案。其思想是基于 LDA 构建了“文本-主题-词”的三层贝叶斯概率模型[3]，即认为每个文本含有多个主题，每个主题内又包含多个关键词。搜索时，首先计算查询关键词与主题的相似度，再根据文档与主题之间的关系匹配符合查询需求的文档。该方法考虑了查询关键词的语义特征，通过主题能够匹配与其语义相近的文档，支持语义搜索。本文的主要贡献如下：

(1)提出了一种支持多关键字的安全语义搜索方法。通过主题模型桥接查询关键字和文档语义关系，并按照相似度得分返回排序结果，实现了基于语义的多关键字安全语义搜索；

(2)提出了一种基于 EMD[4] 结合词嵌入的关键字与主题的语义匹配计算方法，有助于提高查询关键词与文档主题的语义相似度，尤其是当主题中不包含查询关键词的情况；

(3)提出了一种基于主题维度的索引向量生成方法，其索引维度远小于基于关键词的向量维度，不仅改善了生成向量的稀疏问题，还大大减少了索引创建时间和搜索代价。

2　相关工作

2.1　关键字加密可搜索

有关面向单关键字的可搜索加密研究，Song 等[5] 首先提出可搜索加密的概念并实现了可搜索加密方法，但该方法需要逐字扫描整个数据集，搜索时间代价较高；接着，Goh 等[6] 首次为可搜索加密方案提出安全索引，定义了安全模式，并且构造了一个基于伪随机数和布隆过滤器的安全索引；之后，Chang 等[7] 和 Curtmola[8] 进一步完善了安全性定义，并做出了改进。

针对单关键字的加密搜索的局限性，Ballard 等[9] 提出了多关键字的安全搜索方法，但此方法只能查询包含所有关键字的文档，并且无法对结果进行相关性排序；Cao 等[10] 通过生成文档向量和查询向量，实现了多关键字的排序搜索，可返回最相关的 k 个文档；Sun 等[11] 利用 $TF*IDF$ 分数实现带权重的关键字排序搜索，能够提高查询精度。

2.2　关键字加密模糊搜索

在实际搜索中，常常会出现输入拼写错误或者形式不匹配的情况。为此，Li 等[12] 利用通配符实现了关键词模糊搜索，Wang 等[13] 利用索引树提高了搜索效率，但这些模糊搜索方案都需要预先定义字典，会造成额外的空间占用。Wang 等[14] 通过局部敏感哈希和布隆过滤提出了一种新的多关键字模糊搜索方案，但此种方案只针对关键字中一个字母的拼写错误；之后，Fu 等[15] 提出了新的关键字转换方法并考虑了关键字权重，进一步提高了查询精度。

2.3　关键字加密语义搜索

实际上，语义搜索更符合人们的搜索需求，但有关关键字语义加密搜索的研究还很少。Moh 等[16] 提出了基于维基百科和同义词的解决方案，Zhang 等[17] 通过扩展中心关键词语义来达到语义搜索的目的。与已有研究不同，本文没有利用关键词同义语义，而是通过文档主题实现加密语义搜索的解决方案。

3　模型描述

3.1　系统模型

支持加密云数据语义搜索的系统模型与已有模型一样，由数据拥有者、查询用户和公有云服务器 3 部分组成，如图 1 所示。

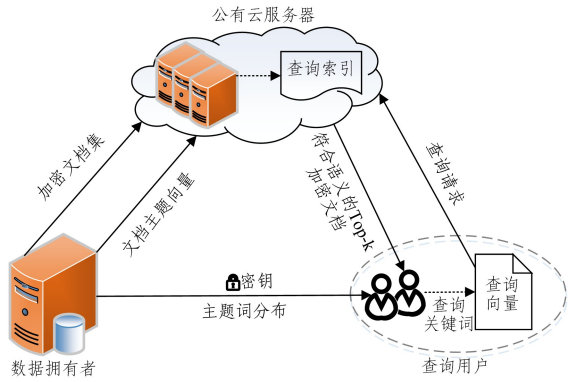


图 1　系统模型

Fig. 1　System model

(1)数据拥有者：明文文档集 $D=(d_1,d_2,\cdots,d_n)$ 中的各个文档生成对应的文档主题向量 $\boldsymbol{W}=(w_1,w_2,\cdots,w_n)$，然后将明文集加密成密文 $C=(c_1,c_2,\cdots,c_n)$，最后将加密后的密文文档集 C 和主题向量 $\boldsymbol{W}$ 一起上传给公有云服务器。

(2)查询用户：用户根据自己的需求输入想要查询的关键字 $K=(k_1,k_2,\cdots,k_n)$，接着根据数据拥有者提供的模型生成查询向量 $\boldsymbol{Q}=(q_1,q_2,\cdots,q_n)$，最后将查询向量提交给公有云服务器完成查询请求，收到返回查询结果后根据密钥 sk 解密得到明文文档。

(3)公有云服务器：主要负责存储数据拥有者的加密文档，并根据接收到的文档主题向量生成索引树，之后再接收用户的查询向量并将查询结果进行排序，返回最相关的 k 篇文档给查询用户。

3.2　模型威胁性分析

本文认为查询用户和数据拥有者是诚实可信的，不会泄露任何有关数据信息，并且严格执行向量生成过程，将结果如实发送给公有云服务器。而公有云服务器则假定是“诚实且好奇”的，具体来说，公有云服务器会诚实并准确地执行索引树建立过程和查询过程等，但同时也可能对数据进行记录分析，采取一定的措施获得额外的信息。本文在如下两种威胁模型下进行分析[10]。

(1)已知密文模型：在此模型中，云服务器只知道加密索引、密文文档集和加密的查询向量。云服务器只能通过密文攻击获取信息。

(2)已知背景模型：在此模型中，云服务器会获得如文档集背景知识或者文档关键词统计等相关信息，攻击者可以基于这些先验知识和对用户所发查询的观察进行推断的被动攻击[18]。

3.3 符号说明

本文中使用的主要符号及其说明如表 1 所列。

表 1 主要符号说明

Table 1 Main symbol description

符号	说明
$F=(f_1,f_2,\cdots,f_m)$	明文文档集合
$C=(c_1,c_2,\cdots,c_m)$	密文文档集合
$T=(t_1,t_2,\cdots,t_k)$	主题集合
$W_k=(w_{k1},w_{k2},\cdots,w_{kM})$	主题 k 所含关键词集合
$Q=(q_1,q_2,\cdots,q_N)$	查询关键词
$\vec{\theta}_f$	文档 f 的主题向量
$\vec{\varphi}_k$	主题 k 的词分布向量
$\vec{\Phi}$	查询向量
$FID=(fid_1,fid_2,\cdots,fid_n)$	文档标识符

4 相关知识

4.1 LDA 主题模型

LDA 主题模型[3]是一个三层的贝叶斯结构，其目的是生成每一篇文档的主体分布和每一个主题中的关键词分布。文档集 F 中的每个文档都对应着一个含 k 个主题的多项分布 θ，可以认为主题代表着文档的语义信息。每个主题又对应着一个含 n 个关键词的多项分布 φ，同时 θ 和 φ 分别拥有一个超参数为 α 和 β 的 Dirichlet 先验分布，即任一文档 f 的主题向量 $\vec{\theta}_f=\text{Dirichlet}(\alpha)$，任一主题 k 的词分布向量 $\vec{\varphi}_k=\text{Dirichlet}(\beta)$。具体的 LDA 模型如图 2 所示。

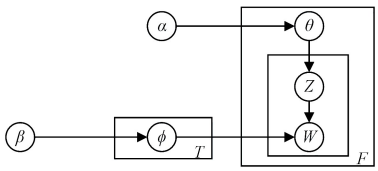


图 2 LDA 模型

Fig. 2 LDA model

4.2 Word2Vec

本文通过 Word2Vec 工具完成关键词的向量化，其基于神经网络从语料库中获得有关词语的语义知识。训练的神经网络模型主要有两种，即 CBOW(Continuous Bag-of-Word Model)和 Skip-gram 模型。CBOW 输入上下文，输出预测词语。Skip-gram 输入当前词语，输出预测上下文，其训练目标优化函数如式(1)所示：

$$L_{\text{Skip-gram}}=\sum_{w_t\in C}\sum_{-k\leqslant j\leqslant k,j\neq 0}\log p(w_{t+j}\mid w_t) \tag{1}$$

其中，w_t 为输入词，k 为上下文大小，C 为训练语料中的所有词。

4.3 EMD 距离

EMD 距离是由运输问题演化而来，具体的模型可以描述为：多个供应商 $P=(p_1,p_2,\cdots,p_m)$，供应量为 $A=(a_1,a_2,\cdots,a_m)$，同时存在着多个经销商 $Q=(q_1,q_2,\cdots,q_n)$，需求量为 $B=(b_1,b_2,\cdots,b_n)$。从 P_i 到 Q_j 的运输成本记为 C_{ij}，A_i 到 B_j 的运输方案记为 F_{ij}，需要找出成本最低的运输方案，则目标函数如式(2)所示：

$$\min(S)=\sum C_{ij}*F_{ij},\forall i\in\{1,2,\cdots,m\},\forall j\in\{1,2,\cdots,n\} \tag{2}$$

利用线性规划可以求出最优 F_{ij}，从而度量一种分布与另一分布之间的相似性。因此，EMD 距离非常适合计算文章主题词分布与查询向量关键词分布之间的语义相似度。

4.4 向量空间模型

向量空间模型用向量表示文档，并且以空间上的相似度表达语义上的相似度，这样，文本之间的分析处理即可以转换为向量之间的运算。文本向量的每一维代表文中的关键字，其值为该关键字在文档中的维度。而用户的查询也被视作同一向量空间下的向量，维度应该与文本向量相同。查询向量与文本向量的内积，即可以表示文档与关键词的相关性。

5 基于 LDA 模型的多关键词语义搜索解决方案

5.1 基于 LDA 的文档-主题-词矩阵生成

数据拥有者为每个文档生成唯一的文档标识符 fid_i 与文档加密密钥 sk，将明文文档集 $F=(f_1,f_2,\cdots,f_m)$ 加密为密文文档集 $C=(c_1,c_2,\cdots,c_m)$，并将 sk 发送给查询用户，将密文文档集上传至公有云服务器。

之后，将明文文档进行分词、去停用词等预处理后，通过 LDA 主题模型得到文档集的主题分布矩阵 $\boldsymbol{\Theta}=\{\vec{f}_f\}_{f=1}^{F}$ 和主题的词分布矩阵 $\boldsymbol{\Phi}=\{\vec{\varphi}_k\}_{t=1}^{T}$。文档主题向量 $\vec{\theta}_f$ 的维度大小为主题个数。主题词分布向量 $\vec{\varphi}_k$ 的维度大小为主题 k 的关键词个数。将文档-主题矩阵传至云服务器，主题-词分布矩阵则发送给查询用户。

5.2 主题向量索引树构建

公有云服务器将所有的主题分布向量构成一个索引树，树节点的数据结构定义为一个四元组 $\langle N_l,N_r,D,FID\rangle$，$N_l$ 为左叶子节点，N_r 为右叶子节点。先对每个文档生成叶子节点，再基于叶子节点生成非叶子节点。主题向量索引树的具体生成过程如算法 1 所示。

算法 1 构建主题向量索引树

输入：文档集主题分布矩阵 $\boldsymbol{\Theta}$

输出：索引树 Γ

1. 初始化队列 CurrentQueue，用来存储树节点。
2. FOR each $f_{FID}\in F$
   生成对应树节点 n；
   n. $D=\vec{\theta}_{FID}$；
   n. N_l=null，n. N_r=null；
   将 n 插入 CurrentQueue。
   END FOR
3. WHILE(队列中节点数目大于 1)
   将 CurrentQueue 头两个节点 n_1 和 n_2 弹出队列，构建新的父节点 pn；
   pn. $N_l=n_1$，pn. $N_r=n_2$；
   pn. D[i]=max(n_1. D[i]，n_2. D[i])；
   把 pn 插入队列。
   END WHILE
4. CurrentQueue 中最后一个节点作为索引树根节点 Γ。
5. RETURN 索引树根节点 Γ。

算法第 1 步初始化节点队列，第 2 步对每个文档都生成对应的节点，第 3－4 步将队列中的节点构建成索引树，最后返回索引树的根节点。

5.3　基于 EMD 结合词嵌入的语义相似度计算

EMD 利用线性规划很好地解决了分布与分布之间的相似性度量问题。本文采取 EMD 距离来计算用户输入的查询关键词和主题之间的语义相似度。在本文中，文档含有 k 个主题 $T=(t_1,t_2,\cdots,t_k)$，每个主题拥有 M 个符合其语义的关键词 $W_k=(w_{k1},w_{k2},\cdots,w_{kM})$和主题词向量$\vec{\varphi}_k$。同时，用户输入 N 个查询关键词 $Q=(q_1,q_2,\cdots,q_N)$。则由式(3)构成 A,B 两个分布：

$$A=\{(\overrightarrow{w_{k1}},\vec{\varphi}_k[1]),(\overrightarrow{w_{k2}},\vec{\varphi}_k[2]),\cdots,(\overrightarrow{w_{kM}},\vec{\varphi}_k[M])\}$$
$$B=\{(\overrightarrow{q_1},1),(\overrightarrow{q_2},1),\cdots,(\overrightarrow{q_N},1)\} \tag{3}$$

其中，$\overrightarrow{w_{ki}}$和$\overrightarrow{q_j}$是关键词经 Word2Vec 生成的词向量，$\vec{\varphi}_k[i]$表示主题词的权重，用户输入的关键词权重设为 1。

$$\min \sum_{i=1}^{M}\sum_{j=1}^{N} d_{ij}s_{ij}$$
$$\text{s.t. } s_{ij}\geqslant 0, i=1,2,\cdots,N; j=1,2,\cdots,M \tag{4}$$
$$\sum_{i=1}^{M} s_{ij}\leqslant \vec{\varphi}_k[i]$$

通过式(4)计算出保证全局代价函数最小的流矩阵 $\boldsymbol{S}=[s_{ij}]$，其中 d_{ij} 表示词向量$\overrightarrow{w_{ki}}$与$\overrightarrow{q_j}$的空间距离。之后可进一步计算 A,B 的 EMD 距离($EMD(A,B)$)，如式(5)所示：

$$EMD(A,B)=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} d_{ij}s_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N} s_{ij}} \tag{5}$$

根据 EMD 距离计算出查询关键词 Q 与主题 W_k 的语义相似度 $Sim(W_k,Q)$，如式(6)所示：

$$Sim(W_k,Q)=EMD(A,B) \tag{6}$$

查询用户输入查询关键词 $Q=(q_1,q_2,\cdots,q_N)$，通过 Word2Vec 将关键词转化为词向量 $\vec{Q}=(\overrightarrow{q_1},\overrightarrow{q_2},\cdots,\overrightarrow{q_N})$。同样，将组成每个主题的关键词转换为词向量$\overrightarrow{W_k}=(\overrightarrow{w_{k1}},\overrightarrow{w_{k2}},\cdots,\overrightarrow{w_{kM}})$。根据式(5)计算查询关键词与主题的 EMD 距离，之后按照式(7)生成查询向量$\vec{\Phi}$，并将其上传至公有云服务器完成进一步的搜索匹配。

$$\overrightarrow{\Phi[i]}=Sim(W_i,Q) \tag{7}$$

5.4　贪婪深度遍历搜索算法

MRSE[10]通常需要遍历所有的索引才能返回符合要求的查询结果，查询效率低下。为了解决此问题，本文采取贪婪深度遍历的方法，返回 top-k 个最符合查询关键词的文档。

在遍历索引树之前，先构建一个含 k 篇文档的集合，其中文档是随机选择的，并计算出文档与查询向量的最小相似度。接着采取深度遍历的方式，若遍历到中间节点，计算该节点与查询向量的内积 I，如果 $I<I_{\min}$，说明此节点下所有的文档都不会比集合内的文档更加相关；如果 $I>I_{\min}$，则继续深度遍历。当遍历到叶子节点时，若 $I>I_{\min}$，则表明叶子节点中的文档更符合查询，所以替换集合中相关度最小的文档，重新计算 $I_{\min}$，继续遍历。最后会留下相关度前 k 的文档，此方法不仅不需要遍历整个索引树，节省了大量的查询时间，还能按相关度对结果进行排序，更加符合用户需求。贪婪深度遍历搜索算法的具体步骤如算法 2 所示。

算法 2　贪婪深度遍历搜索算法

输入：索引树 Γ，查询向量$\vec{\Phi}$

输出：符合查询的 top-k 加密文档

1. 初始化集合 ResultSet；
2. 随机选择 k 篇文档加入 ResultSet，计算其文档主题向量与查询向量内积的最小值 $I_{\min}$；
3. 从根节点开始遍历 DFS(Γ)；
4. IF (node 不是叶子节点)
　　IF 内积大于 $I_{\min}$
　　遍历左孩子节点，DFS(node.N_l)；
　　遍历右孩子节点，DFS(node.N_r)；
　　ELSE
　　RETURN
　　ELSE (node 是叶子结点)
　　IF 内积大于 $I_{\min}$
　　将节点加入集合，同时删除内积为 k_{th} 的节点，重新计算 $I_{\min}$；
　　END IF
5. RETURN ResultSet。

算法第 1 步初始化结果集；第 2 步将随机的 k 个结果加入集合，计算其最小的相似度；第 3－4 步进行深度遍历，并将符合查询的文档加入结果集；最后将结果集返回。

6　安全性分析

(1)文档隐私安全。在本文中，数据拥有者使用对称加密算法对明文文档进行加密，将密钥授予查询用户，公有服务器无法获得密钥，难以破解密文文档，保障了文档的隐私安全。

(2)关键词隐私安全。关键词通过词嵌入的方式转化为词向量，云服务器无法获得查询用户输入的关键词，仅能通过向量内积计算相似度，从而保障了用户输入关键词的隐私安全。

(3)索引隐私安全。索引树节点中含有文档向量、文档标识符。文档向量通过 LDA 模型生成，其过程是不可逆的，所以无法通过文档向量获得文档信息。标识符仅用来识别密文文档，云服务无法通过索引解读文档的具体信息，保障了索引的隐私安全。

(4)查询隐私安全。查询向量通过计算查询关键词与各个主题的 EMD 距离得到，不包含主题内容。云服务器由于不会获得主题的相关信息，因此无法生成有效的陷门，保障了查询的隐私安全。

7　实验验证

本文使用 DBLP 和 20newsgroups 作为测试数据集，实验使用 Python 语言实现。运行环境为 Window 10，处理器主频为 3.40 GHz，内存为 16 GB。数据集描述如表 2 所列。

表 2　数据集描述

Table 2　Dataset description

DataSet	Number of documents
DBLP	15734
20newsgroups	18846

7.1　确定 LDA 主题数

不同的主题数直接影响文档的主题区分和语义识别程度，本文采用困惑度(Perplexity)作为指标选取最佳的主题数。根据在训练集上设置不同的主题数得到的对应 LDA 模型，在测试集上计算困惑度，选择困惑度较小的模型进行后续实验。困惑度计算公式如式(8)所示：

$$perplexity=\exp\left\{-\left(\frac{\sum_{f=1}^{F}\log(p(w))}{\sum_{f=1}^{F}N_f}\right)\right\} \tag{8}$$

$$p(w)=p(t|f)p(w|t)$$

其中，N_f 是文档 f 的单词数，$p(t|f)$ 表示一个文档中每个主题的概率，$p(w|t)$ 表示每个单词在对应主题下出现的概率。

图 3 展示了不同主题数下困惑度的变化情况。可以看出，随着主题数的增加，困惑度呈现明显的下降，当主题数达到一定值后，困惑值趋于稳定。因此，选择 150 为 DBLP 数据集中最合适的主题数，100 为 20newsgroups 数据集中最合适的主题数。

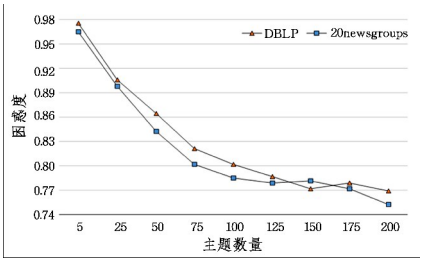
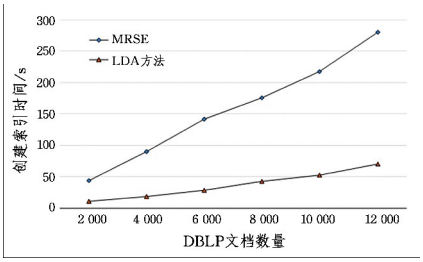


图 3 不同主题数的困惑度

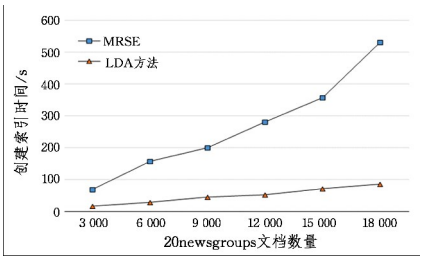
Fig. 3 Perplexity of different topics

7.2 索引生成

图 4 展示了在不同文档数的情况下，传统多关键词排序搜索(Multi-Keyword Ranked Search，MRSE)方法和本文所提的基于 LDA 主题模型方法在不同数据集上创建索引耗时的变化趋势。可以看出，随着文档数的增多，两种方法的索引创建时间都近似呈线性增加。因此，文档数目越多，创建索引所需要的时间也就越多。



(a)DBLP 创建索引的时间



(b)20newsgroups 创建索引的时间

图 4 不同数据集上索引创建时间的对比

Fig. 4 Comparison of index creation time of different data sets

传统 MRSE 方法创建的索引向量维度是由文档的关键词数目决定的，而基于 LDA 方法的索引向量维度仅等于其模型中主题的数目。因此，LDA 的索引远小于传统方法的索引，大幅减少了创建索引的时间。

7.3 陷门生成

陷门即为用户产生的查询向量，向量大小为主题数。查询向量通过 EMD 距离生成，具体生成方法见式(7)。EMD 的时间复杂度由分布大小决定，所以主题包含的关键词数目对陷门生成时间的影响巨大。图 5 给出了主题所含不同数目的关键词对应的陷门生成时间。可以看出，主题所含关键词数目越多，生成陷门的时间越长。但是关键词越多，越能增加主题的区分度，所以需要平衡主题区分度和时间消耗。本文选择主题所含关键词数目为 20 进行后续实验。

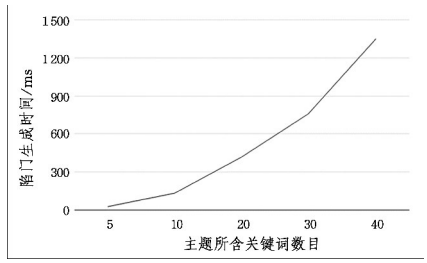
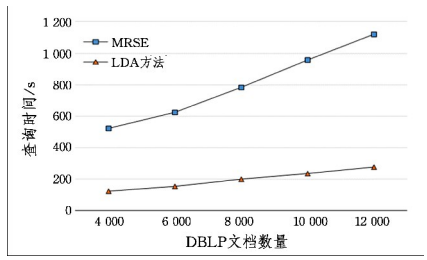


图 5 陷门生成时间

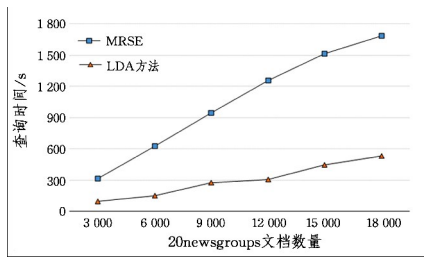
Fig. 5 Trapdoor generation time

7.4 查询时间

图 6 展示了本文方法与传统方法的查询搜索时间开销的对比。当查询文档数增加时，两种方法的查询时间都有所增加，但是传统方法的增加幅度明显大于本方法。这是因为传统方法需要遍历所有索引才能返回结果，本方法由于采用贪婪深度遍历的方法，无须遍历全部索引，因此大大减少了查询所需时间。



(a)DBLP 的查询时间



(b)20newsgroups 的查询时间

图 6 不同数据集上的查询时间的对比

Fig. 6 Comparison of query time of different data sets

7.5 查询结果

首先用 LDA 主题模型提取文档中的主题，各个主题本身就代表着文档的语义信息，并且还不需要任何关于文本的背景知识。另外，根据向量空间模型和 word2vec 理论，关键词所生成的词向量最大程度地保存了该词在原文中的语义信息。最后，EMD 距离很好地度量了关键词与主题词之间的语

义相似度。通过向量内积计算可以对“一词多义”和“一义多词”的语言现象进行建模，这使得搜索系统得到的搜索结果与用户的查询在语义层次上匹配，而不仅是关键词层面上的交集。

本文使用语义拓展度作为语义查询相关性的衡量方法。语义拓展度的定义为：

$$\frac{R(MRSE)\cup R(LDA)}{R(MRSE)} \tag{9}$$

其中，$R(i)$表示方法 i 返回的结果数。

图 7 显示了在不同的返回结果数下语义拓展度的变化情况。

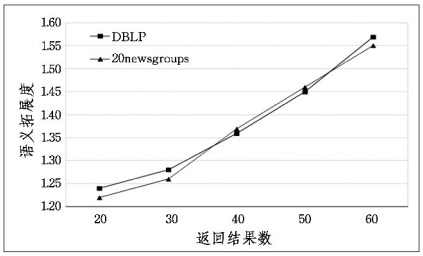


图 7　语义拓展比

Fig. 7　Semantic expansion ratio

可以看出，相比于传统方法，本文所提方法在不同数据集上都很好地拓展了符合语义的查询结果。

结束语　针对传统关键词无法解决语义搜索的问题，提出了一种新型的语义搜索方法，通过 LDA 语义模型桥接关键词和文档之间的语义关系，并极大地减少了查询过程中时间和空间的消耗；另外，通过构建索引树和深度优先的遍历方法，进一步提高了搜索效率。

参 考 文 献

[1] YASUDA M, SHIMOYAMA T, KOGURE J, et al. Secure Pattern Matching Using Somewhat Homomorphic Encryption [C]// Acm Workshop on Cloud Computing Security Workshop. 2013: 65-76.

[2] SUMALATHA N, NAGA S R. A Trusted Hardware Based Database with Privacy and Data Confidentiality[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(3): 752-765.

[3] BLEI D M, NG A Y, JORDAN M I. Latent Dirichlet Allocation [J]. Journal of Machine Learning Research Archive, 2003(3): 993-1022.

[4] RUBNER Y, TOMASI C, GUIBAS L J. The Earth Mover's Distance as a Metric for Image Retrieval[J]. International Journal of Computer, 2000, 40(2): 99-121.

[5] SONG D. Practical Techniques for Searches on Encrypted Data [C]// Proc. of the 2000 IEEE Security and Privacy Symposium. 2000: 44-55.

[6] GOH E J. Building Secure Indexes for Searching Efficiently on Encrypted Compressed data[J]. IACR Cryptology ePrint Archive, 2003, 10(7): 216-234.

[7] CHANG Y C, MITZENMACHER M. Privacy Preserving Keyword Searches on Remote Encrypted Data [C]// International Conference on Applied Cryptography and Network Security. 2004: 442-455.

[8] CURTMOLA R. Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions[C]// ACM Conference on Computer and Communications Security. 2006: 79-88.

[9] BALLARD L, KAMARA S, MONROSE F. Achieving Efficient Conjunctive Keyword Searches over Encrypted Data[C]// 7th International Conference on Information and Communications Security. 2005: 414-426.

[10] CAO N, WANG C, LI M, et al. Privacy-preserving Multi-keyword Ranked Search over Encrypted Cloud Data[C]// 2011 Proceedings IEEE INFOCOM. 2011: 829-837.

[11] SUN W, WANG B, CAO N, et al. Privacy-preserving Multi-keyword Text Search in the Cloud Supporting Similarity-based Ranking[C]// Acm Sigsac Symposium on Information. 2013: 71-82.

[12] LI J, WANG Q, WANG C, et al. Fuzzy Keyword Search over Encrypted Data in Cloud Computingin[C]// IEEE Proc. INFOCOM. 2010: 1-5.

[13] WANG C, REN K, YU S, et al. Achieving Usable and Privacy-assured Similarity Search Over Outsourced Cloud Data[C]// IEEE International Conference on Computer Communications. 2012: 25-30.

[14] WANG B, YU S, LOU W, et al. Privacy-Preserving Multi-Keyword Fuzzy Search over Encrypted Data in the Cloud[C]// IEEE Conference on Computer Communications. 2014: 86-96.

[15] FU Z, WU X, GUAN C, et al. Toward Efficient Multi-Keyword Fuzzy Search Over Encrypted Outsourced Data With Accuracy Improvement[J]. IEEE Transactions on Information Forensics and Security, 2016, 11(12): 2706-2716.

[16] MOH T S, HO K H. Efficient Semantic Search Over Encrypted Data in Cloud Computing[C]// International Conference on High Performance Computing & Simulation. 2014: 382-390.

[17] ZHANG J F, WU X L, WANG Q, et al. Enabling Central Keyword-Based Semantic Extension Search Over Encrypted Outsourced Data[C]// IEEE Transactions on Information Forensics and Security. 2017: 2986-2997.

[18] NING J, XU J, LIANG K, et al. Passive Attacks Against Searchable Encryption[J]. IEEE Transactions on Information Forensics and Security, 2019, 14(3): 789-802.

LI Yan, born in 1995, postgraduate. His main research interests include semantic search and query processing.

SHEN De-rong, born in 1964, professor, Ph.D, supervisor, is a senior member of China Computer Federation. Her research interests include Web data processing and distributed database.