DOI: 10.19783/j.cnki.pspc.230652

基于 MapReduce 的输电监测数据智能检索模型

赵松燕[1]，曲朝阳[1]，郭晓利[1]，余 通[3]，黎 新[2]，谢 铭[1,2]，余 福[4]

(1. 东北电力大学计算机学院，吉林 吉林 132012；2. 广西电网有限责任公司电力科学研究院，广西 南宁 530023；
3. 中国人民银行青海省分行，西宁 青海 810000；4. 贵州电网有限责任公司毕节供电局，贵州 毕节 551700)

摘要：随着新型电力系统发展，输电监测文本数据呈现出体量大、增速快等特点，且因行业数据传输协议私有化，导致数据检索性能低，影响输电线路实时决策分析。因此提出了基于 MapReduce 的输电监测数据智能检索模型。首先，改进了 SimHash 算法，实现输电线路在线监测文本数据检索向量的高效提取。并引入多属性决策以及综合评分机制，实现目标数据的精准检索，提升数据的检索精度及查全率。其次，针对数据体量大、增速快的特点，设计了基于 MapReduce 的电力数据检索模型。最后，通过电网实例对比分析，验证了所提方法的检索精度、查全率及检索效率。

关键词：新型电力系统；输电线路数据；改进 SimHash；智能检索；MapReduce

Intelligent retrieval model of power transmission monitoring data based on MapReduce

ZHAO Songyan[1], QU Zhaoyang[1], GUO Xiaoli[1], YU Tong[3], LI Xin[2], XIE Ming[1,2], YU Fu[4]

(1. School of Computer Science, Northeast Electric Power University, Jilin 132012, China; 2. China Southern Power Grid Guangxi Power Grid Co., Ltd. Research Institute, Nanning 530023, China; 3. People's Bank of China Qinghai Provincial Branch, Xining 810000, China; 4. Bijie Power Supply Bureau, China Guizhou Power Grid Co., Ltd. Bijie 551700, China)

Abstract: With the development of new power systems, transmission monitoring text data presents the characteristics of large volume and fast growth rate, and because of the privatization of industry data transmission protocols, the performance of data retrieval is low. This affects the real-time decision analysis of transmission lines. An intelligent retrieval model of power transmission monitoring data based on MapReduce is proposed. First, this paper innovates and improves the SimHash algorithm to achieve efficient extraction of retrieval vectors for transmission line online monitoring text data, and introduces multi-attribute decision-making and comprehensive scoring mechanisms to achieve precise retrieval of target data and improve data retrieval accuracy and recall. Second, from the characteristics of large data volume and fast growth rate, a power data retrieval model based on MapReduce is designed. Finally, the retrieval accuracy, recall rate and retrieval efficiency of the proposed method are verified through comparative analysis of power grid examples.

This work is supported by the National Natural Science Foundation of China (No. 6217111).

Key words: new power systems; transmission line data; improve SimHash; intelligent retrieval; MapReduce

0 引言

新型电力系统是实现“碳达峰、碳中和”的主要方式，具有高比例电力电子设备接入的特征，而输电线路监测系统是其应用的关键基础设施，是电力监测数据的主要来源，如故障常常会对电网输电线路的“健康”状态的监测产生不良影响[1-3]。

然而，在工程实践应用中，随着新型电力系统的建设，大量电力电子设备等异构终端接入电网，这使得电网输电线路在线监测数据交互多元化，并呈现出体量大、增速快、传输协议私有以及分布式存储等特点[4-6]。同时，由于现有的输电线路检索系统主要基于关键词顺序查找算法设计，在数据规模

基金项目：国家自然科学基金项目资助(6217111)；吉林省科技发展计划项目资助(20210203195SF)；南网广西电网公司科技项目资助(GXKJXM20222017)；南网广西电网公司创新项目资助(047000KK52210036)

体量大、增速快的环境下，往往存在数据检索不够及时、精度低、有效性不足等问题，常导致检索结果可用率低，难以有效支撑输电线路的故障定位需求，进而往往难以有效支撑对电网输电线路的“健康”状态的监测。

当前对解决信息智能检索的问题的研究主要集中在本体法、语义法等方面。文献[7]针对多主题编码文档的查询中，单表示模型可能会造成严重文档信息丢失的问题，设计了一种新的语义检索方法，实现了不同主题文档信息的识别。文献[8]针对电网调控多维信息缺乏有效检索手段，提出面向电网调控信息智能检索的知识图谱构建方法，具有较高的识别准确率，能够支撑不同场景下调控信息的智能检索。文献[9]引入麻雀搜索算法来优化数据检索的鲁棒问题。文献[10]针对现有的自动语音识别技术在商业智能检索系统中，远程交互性差、检索覆盖面低以及智能化不高的问题。提出了一种互操作性强、智能高效的解决方案，有效提升商业智能中的数据检索效率。文献[11]提出了检索增强生成模型，提升了多源数据识别的精度和效率。文献[12]针对密集检索问题中时间复杂度高的问题，提出了密集检索工具包，实现了文本处理、模型训练、语料库查询编码和搜索。文献[13]针对云环境下的信息安全问题，借助多阶段身份验证和优化的河豚算法开发了高效的安全数据检索模型，解决了云用户多重身份验证问题。文献[14]针对大规模数据集环境下，数据缺失或未标记的新领的数据域训练问题，提出了基于 K 最近邻算法的大规模数据检索思路，提升了数据的检索精度。文献[15]针对现有研究大多忽视了软件缺陷报告所属的版本与目标源代码的版本之间存在的“版本失配”问题或在训练和测试模型时缺陷报告的时间顺序所引发的“数据泄露”问题，分析了基于信息检索的缺陷定位模型的优劣。这些方法虽然在数据缺失处理、商业智能、文本训练以及云安全检索和软件缺陷报告定位等方面取得了一定的效果，但忽略了新型电力系统背景下输电线路数据分布式存储，以及数据存储结构和行业传输协议私有等特点，不具有电力行业数据检索的实用性，难以满足当下电力行业特定场景下的检索需求。

为此，为解决现有输电线路故障检索系统中，已有的故障数据集的有效检索问题，本文提出了基于 MapReduce[16]的输电线路在线监测数据智能检索模型。其中，改进 SimHash 算法并引入多属性决策技术及综合评分机制，使特征关键词的提取变得更加快速、准确，目标筛选更具实用性，实现了大规模输电线路在线监测数据的高效、精准及较为全面的检索。

1 输电线路在线监测数据检索模型

输电线路在线监测数据检索模型是基于 MapReduce 的编程模型，具有应用于电网某电力公司的实际效果。本文改进 SimHash 算法并引入多属性决策技术，设计了基于 MapReduce 的输电监测数据智能检索模型(MapReduce-intelligent retrieval method, MR-IRM)。MR-IRM 结合了 SimHash 算法[17]、MapReduce 模型、词频−逆向文件频率(term frequency-inverse document frequency, TF-IDF)和多属性决策等方法，有效实现输电线路在线监测数据的并行检索。在模型中，Map 函数可同时访问多个数据分片，且结合检索需求，提取检索元组的行号及其属性值，然后进行映射计算，同时调用相关函数，生成＜key, value＞键值对；Reduce 函数接收各个节点 Map 函数的输出结果，进行归并计算，并将结果存储在分布式文件系统(hadoop distributed file system, HDFS)中，MR-IRM 原理如图 1 所示。

由图 1 可以看出：模型包括 6 个过程，其中含 9 个阶段。过程 1 包括两个阶段，即构造关键词数据库 D_k 、关键词检索向量 $\boldsymbol{G}_K$ ；过程 2 包括两个阶段，即构造序列检索数据库向量 D_{k+1} 和序列检索向量 $\boldsymbol{X}_K$ ；过程 3 包括两个阶段，即构造检索特征数据库向量 $\boldsymbol{D}_{k+1}$ 和检索特征向量 $\boldsymbol{R}_K$ ；过程 4 包括一个阶段，即构建相似度检索向量 $\boldsymbol{V}_K$ ；过程 5 包括一个阶段，即构建效用值检索向量 $\boldsymbol{T}_K$ ；过程 6 包括一个阶段，即构建综合检索向量 $\boldsymbol{Z}_K$ 。模型的 Step2 和 Step3 是基于 MapReduc 模型，并在引入多属性决策和综合评分机制的基础上完成对 Step1 的并行建模，最终实现数据检索模型的构建，主要体现在过程 5 和过程 6 上。

2 基于改进 SimHash 的检索特征向量提取

数据检索时须实现特征向量的提取，且该特征向量以二进制串形式存储。传统 SimHash 算法可实现特征向量的提取以及文本数据到二进制串的转换，但存在准确率较低、效率不高、权重计算精度缺失等不足。因此，针对数据特征向量的提取问题，设计了基于改进 SimHash 的检索特征向量提取策略，步骤如下。

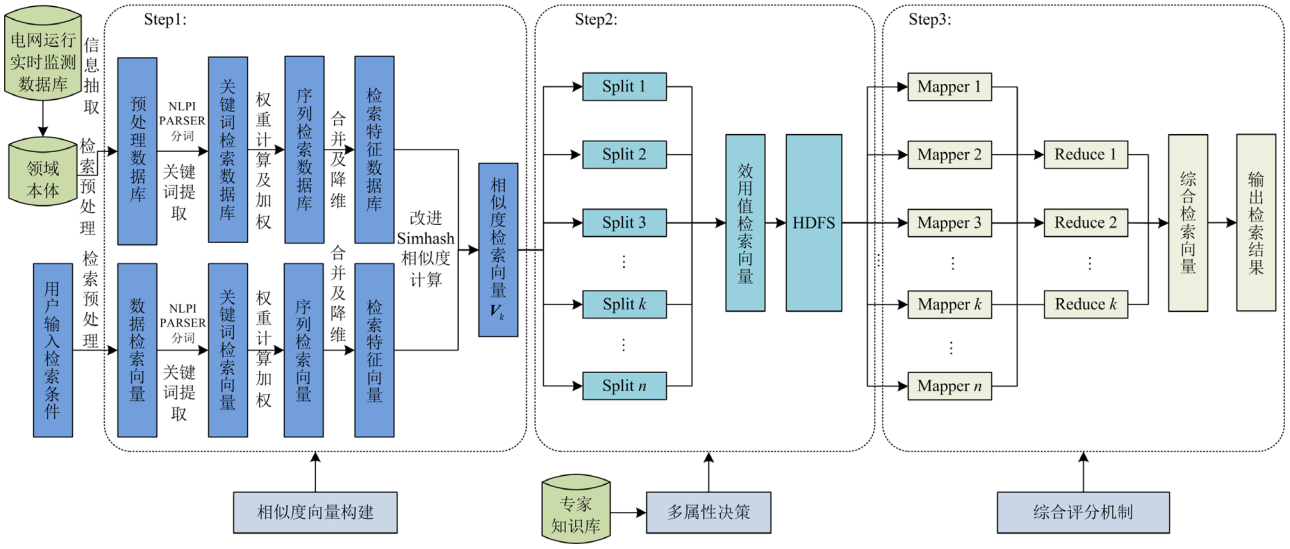

图 1　MR-IRM 算法智能检索模型

Fig. 1 Intelligent retrieval model of MR-IRM algorithm

Step1：关键词提取。采用 NLPIR PARSER 分词工具进行中文分词，完成新词发现、自适应分词以及关键词提取[18]，输出关键词 $t_{(i,n)}$，$t_{(i,n)}$ 表示第 i 个元组中的第 n 个关键词。

Step2：特征向量提取及权重计算。采用词性(其中名词权重为 0.3，动词权重设为 0.2 等[19])、词频、新词、词跨度权重相结合的权重计算方式对关键词的特征向量进行提取。关键词的词性权重通过层次分析法所得，词性权重为 $w_{1(i,j)}$，表示第 i 个元组的第 j 个关键词的词性权重。

Step3：采用词频−逆向文件频率 TF-IDF[20]算法计算关键词 $t_{(i,n)}$ 的词频权重 $w_{2(i,j)}$，表示第 i 个元组的第 j 个关键词的词频权重，并加入新词权重 $w_{3(i,j)}$ 及词跨度权重 $w_{4(i,j)}$，其中 $w_{3(i,j)}$ 表示第 i 个元组中第 j 个新词的词频权重，$w_{4(i,j)}$ 表示第 i 个元组中第 j 个关键词的词跨度权重。最后得出关键词 $t_{(i,n)}$ 的权重 $W_{(i,n)}$，即 $W_{(i,n)}$ 是第 i 个元组中的第 j 个关键词 $t_{(i,n)}$ 的权重，最后词性权重表 $Q=\{W_{(i,n)}\}$。

1) 关键词频率(term frequency, TF)

TF 表示在检索数据库中出现某个关键词的频率。关键词 $t_{(i,n)}$ 的关键词频率可定义为

$$T_{\mathrm{F}i,j}=\frac{n_{i,j}}{\sum\limits_{Q}n_{i,j}} \tag{1}$$

式中：$n_{i,j}$ 是第 i 个元组中的第 j 个关键词 $t_{(i,n)}$ 在检索数据库中出现的频数；$\sum\limits_{Q}n_{i,j}$ 是检索数据库中全部词出现总数。

2) 逆文档频率(inverse document frequency, IDF)

IDF 用于评定某关键词是否重要的准则，因此对于关键词的 IDF 的定义，如式(2)所示。

$$I_{\mathrm{DF},i}=\lg(N/n_{i,j}+\alpha) \tag{2}$$

式中：N 是数据库中词的总数；α 为一个经验值，一般取 0.01、0.1、1。

3) 传统 TF-IDF

(1) 求关键词 $t_{(i,j)}$ 的词频权重 $w_{2(i,j)}$

$$w_{2(i,j)}=T_{\mathrm{F}\,i,j}\times I_{\mathrm{DF},i}=\frac{n_{i,j}}{\sum\limits_{Q}n_{i,j}}\times\lg\left(\frac{N}{n_{i,j}}+a\right) \tag{3}$$

(2) 求关键字 $t_{(i,n)}$ 的新词权重 $w_{3(i,j)}$

$$w_{3(i,j)}=\frac{\dfrac{n_{i,j}}{\sum\limits_{Q}n_{i,j}}\times\lg\left(\dfrac{N}{n_{i,j}}+\alpha\right)+L_{\mathrm{en}}(t)}{\sqrt{\sum\limits_{p=1}^{j}\left[\dfrac{n_{i,j}}{\sum\limits_{Q}n_{i,j}}\times\lg\left(\dfrac{N}{n_{i,j}}+\alpha\right)\right]^2+L_{\mathrm{en}}(t)}} \tag{4}$$

式中，$L_{\mathrm{en}}(t)$ 是新词的长度。

(3) 关键字 $t_{(i,n)}$ 的词跨度权重 $w_{4(i,j)}$

$$w_{4(i,j)}=\frac{l_i}{L} \tag{5}$$

式中：l_i 是 $t_{(i,n)}$ 所在的段数；L 是段落总数。

4) 改进 TF-IDF

关键词 $t_{(i,n)}$ 的权重 $W_{(i,n)}$ 定义为

$$W_{(i.j)}=w_{1_{(i.j)}}\times w_{2_{(i.j)}}\times w_{3_{(i.j)}}\times w_{4_{(i.j)}}=$$
$$w_{1_{(i.j)}}\times[\frac{n_{i,j}}{\sum_{Q}n_{i,j}}\times\lg(\frac{N}{n_{i,j}}+a)]\times$$
$$\frac{\frac{n_{i,j}}{\sum_{Q}n_{i,j}}\times\lg(\frac{N}{n_{i,j}}+a)+L_{\mathrm{en}}(t)}{\sqrt{\sum_{p=1}^{j}[\frac{n_{i,j}}{\sum_{Q}n_{i,j}}\times\lg(\frac{N}{n_{i,j}}+a)]^2+L_{\mathrm{en}}(t)}}\times\frac{l_i}{L} \tag{6}$$

3 基于多属性决策的目标筛选

由于相似度检索中存在信息过载问题，为了对检索结果进行有导向性筛选，在改进 SimHash 的基础之上，引入了基于多属性决策方法[21-22]来计算检索结果的效用值，以达到快速检索目标的目的，其基本思想如下所述。

借助多属性决策方法“理想解”和“负理想解”的思路来逐一排序，再判定各个检索方案的好坏[23]。在 k 个检索方案中，如果其中一个的检索结果与理想解的距离相对较近，并且与负理想解的距离最远，那么它就是 k 个检索方案中检索到的最优解。其中预置了专家知识，对不同时间和环境的数据从多个因素(如天气、高峰期等)作出客观评分，作为理想解的指标。在检索过程中，基于多属性决策方法的比较和评价过程如下所述。

1) 将数据属性特征作为检索结果的评价指标，创建初始决策矩阵 $\boldsymbol{F}=\boldsymbol{V}_k\times\boldsymbol{B}_k$。

设检索到的数据相似度检索向量 $\boldsymbol{V}_k=[v_1,\cdots,v_k]$，$v_k=v_{i,j}$，$v_{i,j}$ 是第 i 个元组的第 j 个属性值。设检索需求指标向量 $\boldsymbol{B}_k=[b_1,\cdots,b_k]$，其中 $b_1,\cdots,b_k$ 可分别表示人工经验分析、辅助决策手段、天气等多个属性的权重，$b_k=b_{i,j}$，$b_{i,j}$ 是第 i 个元组的第 j 个属性值的检索需求量。因此，初始决策矩阵 $\boldsymbol{F}=[\boldsymbol{V}_k\times\boldsymbol{B}_k]=[(v_{i,j}\times b_{i,j})]_{i\times j}$，其中 $v_{i,j}\times b_{i,j}$ 表示在第 i 个数据中，它的第 j 个指标的分值。

2) 对初始决策矩阵进行规范化得到规范化决策矩阵 $\boldsymbol{F}^*$，$\boldsymbol{F}^*=[V_k^*\times F_k^*]=[(v_{i,j}^*\times f_{i,j}^*)]_{i\times j}$。

$$v_{i,j}^*\times f_{i,j}^*=v_{i,j}\times f_{i,j}\Big/\sum_{i=1}^{k}(v_{i,j}\times f_{i,j})^2 \tag{7}$$

3) 将检索需求指标向量 $\boldsymbol{B}_k=[b_1,\cdots,b_k]$ 进行规范化，得到规范化后的需求向量 $\boldsymbol{B}_k^*=[b_1^*,\cdots,b_j^*]$，其中 b_j^* 表示规范化后的第 j 个需求指标的权重，故

$$b_j^*=b_j\Big/\sum_{j=1}^{n}b_j \tag{8}$$

4) 分别计算每个输电线路在线监测检索数据 $P_i=(1\leqslant i\leqslant k)$ 到理想解 φ_i^+ 和负理想解 φ_i^- 的距离。

$$\varphi_i^+=\sqrt{\sum_{j=1}^{n}[b_j^*(v_{i,j}^*\times f_{i,j}^*-vf_{j\_\mathrm{best}}^*)]^2} \tag{9}$$

$$\varphi_i^-=\sqrt{\sum_{j=1}^{n}[b_j^*(v_{i,j}^*\times f_{i,j}^*-vf_{j\_\mathrm{worst}}^*)]^2} \tag{10}$$

式中：n 是输电线路在线监测数据关于定性特征的检索总量；$(v_{i,j}^*\times f_{i,j}^*)$ 表示对 $(v_{i,j}\times b_{i,j})$ 规范化后的评分值；$vf_{j\_\mathrm{best}}^*$、$vf_{j\_\mathrm{worst}}^*$ 是独立的变量，分别表示规范化后得到的理想解的第 j 个子评分值和所得的负理想解的第 j 个子评分值。

5) 计算所有 $P_i=\ (1\leqslant i\leqslant k)$ 的效用值 U_i 为

$$U_i=\frac{\varphi_i^-}{\varphi_i^++\varphi_i^-} \tag{11}$$

4 MR-IRM 智能检索模型实现

MR-IRM 智能检索模型主要分为 6 个过程，其中包括九个阶段，实现步骤如下所述。

先设置两个条件：(1) 设构造关键词数据库为 D_k，$k\in R$，行号记为 ID，数据库的第 i 行 j 列的属性值为 $v_{i,j}$，且 $v_{i,j}\in V$，$t_{(i,n)}$ 是第 i 行第 n 个关键词；(2) 设用户输入的检索条件数据表为 Y_k，$k\in R$，行号记为 id，数据表的第 i 行 j 列的属性值记为 $E_{i,j}$ 且 $E_{i,j}\in E$，$m_{(i,n)}$ 表示用户输入的第 i 行第 n 个关键词。MR-IRM 模型的实现过程如下。

过程 1：构造关键词数据库 D_k 及关键词检索向量 $\boldsymbol{G}_k$。

输入：D_k 和 Y_k；

输出：D_k 对应的关键词检索数据库向量 $\boldsymbol{D}_i$、Y_k 对应的关键词检索向量 $\boldsymbol{G}_i$。

1) 算法 1：Map 部分

(1) 输入 $\langle\mathrm{key}=ID_i,\mathrm{value}=v_{i,j}\rangle$、$\langle\mathrm{key}=id_i,\mathrm{value}=E_{i,j}\rangle$。

(2) 基于 NLPIR PARSER 中文分词工具对每个 $v_{i,j}$ 和 $E_{i,j}$ 进行分词，将 $v_{i,j}$ 和 $E_{i,j}$ 拆分为不同的关键词 $t_{(i,n)}$ 和 $m_{(i,n)}$；按照属性值逐一进行排序，以 $t_{(i,n)}$ 和 $m_{(i,n)}$ 为元素构造 $\boldsymbol{D}_i=[t_{(i,n)}]$ 和 $\boldsymbol{G}_i=[m_{(i,n)}]$。

(3) 输出键值对 $\langle\mathrm{key}=ID_i,\mathrm{value}=\boldsymbol{D}_i\rangle$ 和 $\langle\mathrm{key}=$

$id_i, \text{value} = \boldsymbol{G}_i\rangle$，并存储于 HDFS 中。

2) 算法 2：Reduce 部分

(1) 输入算法 1 中 Map 的输出结果。

(2) while：$i = 1,2,\cdots,size(D_k)$；do：根据键值 $\boldsymbol{D}_i$ 归并计算；while：$i = 1,2,\cdots,size(Y_k)$；do：根据键值 $\boldsymbol{G}_i$ 归并计算。其中，$\text{size}(D_k)$ 是 D_k 中所有的被检索元组数量，$\text{size}(Y_k)$ 是 Y_k 中所有的待检索元组数量。

(3) 输出 $\langle \text{key} = \boldsymbol{D}_i/\cdots, \text{value} = (ID_1,\cdots/\cdots)\rangle$、$\langle \text{key} = \boldsymbol{G}_i/\cdots, \text{value} = (id_1,\cdots/\cdots)\rangle$。

将输出的结果以关键词检索数据库子向量 $\boldsymbol{d}_k = (ID_i, \boldsymbol{D}_i)$、关键词检索子向量 $\boldsymbol{g}_k = (id_i, \boldsymbol{G}_i)$ 的形式存储在 HDFS 中，如果一个 key 对应多值 value，则将 value 拆分，且分别与 key 一一对应，组织为不同的检索向量并近邻地存放在 $\boldsymbol{d}_k$ 和 $\boldsymbol{g}_k$ 中；在 $\boldsymbol{d}_k$、$\boldsymbol{g}_k$ 中具有相同 key 的检索均分别被聚类到同一向量中，故得到 $\boldsymbol{D}_K = [\boldsymbol{d}_k]$、$\boldsymbol{G}_k = [\boldsymbol{g}_k]$；

过程 2：构造序列检索数据库向量 $\boldsymbol{D}_{k+1}$ 及序列检索向量 $\boldsymbol{X}_K$。

输入：关键词检索数据库向量 $\boldsymbol{d}_k$、关键词检索向量 $\boldsymbol{g}_k$ 和词性权重表 Q；

输出：检索数据库向量 $\boldsymbol{D}_{k+1}$ 和检索向量 $\boldsymbol{X}_K$。

3) 算法 3：Map 部分

(1) 读取 HDFS 文件中的 $\boldsymbol{d}_k$、$\boldsymbol{g}_k$ 和算法结果 $\langle \text{key} = ID_i, \text{value} = \boldsymbol{D}_i\rangle$、$\langle \text{key} = id_i, \text{value} = \boldsymbol{G}_i\rangle$；

(2) 对每一个 $t_{(i,n)} \in \boldsymbol{D}_i$，按照 $t_{(i,n)}$ 的词性设置对应词性权重 $w_{1(i,n)}$、词频权重 $w_{2(i,n)}$、新词权重 $w_{3(i,n)}$ 及词跨度权重 $w_{4(i,n)}$。同理，对 $m_{(i,n)} \in \boldsymbol{G}_i$，按照 $m_{(i,n)}$ 的词性以相同方式计算对应的词性权重 $w^*_{1(i,n)}$、词频权重 $w^*_{2(i,n)}$、新词权重 $w^*_{3(i,n)}$ 及词跨度权重 $w^*_{4(i,n)}$。并按以下来矩阵格式存储，即

$\boldsymbol{w}_{1i} = [w_{1(i,n)}]$、$\boldsymbol{W}_1 = [\boldsymbol{w}_{1i}]$，$\boldsymbol{w}_{2i} = [w_{2(i,n)}]$、$\boldsymbol{W}_2 = [\boldsymbol{w}_{2i}]$，$\boldsymbol{w}_{3i} = [w_{3(i,n)}]$、$\boldsymbol{W}_3 = [\boldsymbol{w}_{3i}]$；$w_{4(i,n)} = T_{\text{F}} - I_{\text{DF}}(t_{(i,n)})$，$\boldsymbol{w}_{4i} = [w_{4(i,n)}]$、$\boldsymbol{W}_4 = [\boldsymbol{w}_{4i}]$，$\boldsymbol{w}^*_{1i} = [w^*_{1(i,n)}]$、$\boldsymbol{W}^*_1 = [\boldsymbol{w}^*_{1i}]$，$\boldsymbol{w}^*_{2i} = [w^*_{2(i,n)}]$、$\boldsymbol{W}^*_2 = [\boldsymbol{w}^*_{2i}]$，$\boldsymbol{w}^*_{3i} = [w^*_{3(i,n)}]$、$\boldsymbol{W}^*_3 = [\boldsymbol{w}^*_{3i}]$，$w^*_{4(i,n)} = T_{\text{F}} - I_{\text{DF}}(m_{(i,n)})$、$\boldsymbol{w}^*_{4i} = [w^*_{4(i,n)}]$，$\boldsymbol{W}^*_4 = [\boldsymbol{w}^*_{4i}]$。

(3) 输出 $\langle \text{key} = ID, \text{value} = \boldsymbol{W}_1\rangle, \cdots, \langle \text{key} = ID, \text{value} = \boldsymbol{W}_4\rangle$；

$\langle \text{key} = id, \text{value} = \boldsymbol{W}^*_1\rangle, \cdots, \langle \text{key} = id, \text{value} = \boldsymbol{W}^*_4\rangle$

(4) $i = 1,2,\cdots,size(D_k)$、$j = 1,2,\cdots,n$；$i = 1,2,\cdots,size(Y_k)$、$j = 1,2,\cdots,n$。

While：

$(\langle \text{key} = ID, \text{value} = \boldsymbol{W}_1\rangle, \cdots, \langle \text{key} = ID, \text{value} = \boldsymbol{W}_4\rangle, \langle \text{key} = id, \text{value} = \boldsymbol{W}^*_1\rangle, \cdots, \langle \text{key} = id, \text{value} = \boldsymbol{W}^*_4\rangle)$。

do：

$W_{(i,n)} = w_{1(i,n)} \times w_{2(i,n)} \times w_{3(i,n)} \times w_{4(i,n)}$；

$\boldsymbol{w}_i = (w_{(i,n)}) = [w_{(i,1)}, w_{(i,2)}, \cdots, w_{(i,n)}]$；

$\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_i]$；

$W^*_{(i,n)} = w^*_{1(i,n)} \times w^*_{2(i,n)} \times w^*_{3(i,n)} \times w^*_{4(i,n)}$；

$\boldsymbol{w}^*_i = (W^*_{(i,n)}) = [W^*_{(i,1)}, W^*_{(i,1)}, \cdots, W^*_{(i,n)}]$；

$\boldsymbol{W}^* = [\boldsymbol{w}^*_1, \boldsymbol{w}^*_2, \cdots, \boldsymbol{w}^*_n]$；

(5) 输出 $\langle \text{key} = ID, \text{value} = \boldsymbol{W}\rangle$、$\langle \text{key} = id, \text{value} = \boldsymbol{W}^*\rangle$。

4) 算法 4：Map 部分

(1) 输入算法 2 中的结果 $\langle \text{key} = ID_i, \text{value} = \boldsymbol{D}_i\rangle$、$\langle \text{key} = id_i, \text{value} = \boldsymbol{G}_i\rangle$。

(2) 当 $i = 1,2,\cdots,size(D_k)$、$j = 1,2,\cdots,n$；$i = 1,2,\cdots,size(Y_k)$、$j = 1,2,\cdots,n$)；$f_{(i,n)} = \text{Hash}(t_{(i,n)})$，$f^*_{(i,n)} = \text{Hash}(m_{(i,n)})$ 时，$f_{(i,n)}$ 是关键词 $t_{(i,n)}$ 哈希值且为 r 维，$f^*_{(i,n)}$ 是关键词 $m_{(i,n)}$ 哈希值且为 r 维，并按以下矩阵格式存储，即：$\boldsymbol{f}_i = [f_{(i,1)}, f_{(i,2)}, \cdots, f_{(i,n)}]$；$\boldsymbol{f} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_i]$；$\boldsymbol{f}^*_i = [f^*_{(i,1)}, f^*_{(i,2)}, \cdots, f^*_{(i,n)}]$；$\boldsymbol{f}^* = [\boldsymbol{f}^*_1, \boldsymbol{f}^*_2, \cdots, \boldsymbol{f}^*_i]$；用相同的传统 Hash 函数对 $t_{(i,n)}$、$m_{(i,n)}$ 求 Hash 值，并定长 r 维输出。

(3) 输出 $\langle \text{key} = ID, \text{value} = \boldsymbol{f}\rangle$、$\langle \text{key} = id, \text{value} = \boldsymbol{f}^*\rangle$。

5) 算法 5：Map 部分

(1) 输入算法 3 结果中的 $\langle \text{key} = ID, \text{value} = \boldsymbol{W}\rangle$、$\langle \text{key} = id, \text{value} = \boldsymbol{W}^*\rangle$ 和 $\langle \text{key} = ID, \text{value} = \boldsymbol{f}\rangle$、$\langle \text{key} = id, \text{value} = \boldsymbol{f}^*\rangle$。

(2) 初始化 r 维向量 $\boldsymbol{F}_{(i,n)} = (0)$、$\boldsymbol{F}^*_{(i,n)} = (0)$；$\boldsymbol{S} = (0)$、$\boldsymbol{S}^* = (0)$。当 $i = 1,2,\cdots,size(D_k)$、$j = 1,2,\cdots,n$；$i = 1,2,\cdots,size(Y_k)$、$j = 1,2,\cdots,n$；且 $\max(\lambda) = r$ 时，若 $f_{(i,n)}$、$f^*_{(i,n)}$ 的第 λ 位分别大于 0，则分别将 $\boldsymbol{F}_{(i,n)}$、$\boldsymbol{F}^*_{(i,n)}$ 的第 λ 位分别加上 $w_{1(i,n)}$、$w^*_{1(i,n)}$；否则，将其减去 $w_{1(i,n)}$、$w^*_{1(i,n)}$，并按以下矩阵格式存储，即 $\boldsymbol{F}_i = [\boldsymbol{F}_{(i,n)}]$；$\boldsymbol{F} = [\boldsymbol{F}_i]$；$\boldsymbol{F}^*_i = [\boldsymbol{F}^*_{(i,n)}]$；$\boldsymbol{F}^* = [\boldsymbol{F}^*_1, \cdots, \boldsymbol{F}^*_i]$。

(3) 当 $i=1,2,\cdots,size(D_k)$、$j=1,2,\cdots,n$；$i=1,2,\cdots,size(Y_k)$、$j=1,2,\cdots,n$ 时，执行 $\boldsymbol{S}_i=\Delta F_{i,j}$，$\boldsymbol{S}=[\boldsymbol{S}_i]$；$\boldsymbol{S}_i^*=\Delta F_{i,j}^*$，$\boldsymbol{S}^*=[\boldsymbol{S}_i^*]$，其中 $\Delta F_{i,j}$、$\Delta F_{i,j}^*$ 分别指具有相同维度的不同字符串 $F_{i,j}$、$F_{i,j}^*$ 的对应位求和。

(4) 输出 $\langle \text{key}=ID, \text{value}=\boldsymbol{S}\rangle$、$\langle \text{key}=id, \text{value}=\boldsymbol{S}^*\rangle$。

6) 算法 6：Reduce 部分

(1) 输入 Map 的结果 $\langle \text{key}=ID, \text{value}=\boldsymbol{S}\rangle$、$\langle \text{key}=id, \text{value}=\boldsymbol{S}^*\rangle$。

(2) 以 $\boldsymbol{S}$ 为 key 进行归并计算。

(3) 输出 $\langle \text{key}=\boldsymbol{S}_i/\cdots, \text{value}=(ID_1,\cdots/\cdots)\rangle$、$\langle \text{key}=\boldsymbol{S}_i^*/\cdots, \text{value}=(id_1,\cdots/\cdots)\rangle$；并以序列检索数据库子向量 $\boldsymbol{d}_{k+1}=(ID_i,\boldsymbol{S}_i)$、序列检索子向量 $\boldsymbol{x}_{k+1}=(ID_i,\boldsymbol{S}_i^*)$ 的形式存储在 HDFS 中，如果一个 key 对应多值 value，则将 value 拆分，且分别与 key 一一对应，组织为不同的元组，并分别按 ID、id 从大到小存放在序列检索数据库子向量 $\boldsymbol{d}_{k+1}$、序列检索子向量 $\boldsymbol{x}_k$ 中，且 $\boldsymbol{D}_{k+1}=[\boldsymbol{d}_{k+1}]$、$\boldsymbol{X}_k=[\boldsymbol{x}_k]$。

过程 3：构造检索特征数据库向量 $\boldsymbol{D}_{k+2}$ 及检索特征向量 $\boldsymbol{R}_K$。

输入：序列检索数据库向量 $\boldsymbol{D}_{k+2}$、检索向量 $\boldsymbol{x}_k$；

输出：$\boldsymbol{D}_{k+2}$、$\boldsymbol{R}_K$。

7) 算法 7：Map 部分

(1) 初始化 r 维向量 $\boldsymbol{S}_i=(0)$、$\boldsymbol{S}_i^*=(0)$；读取算法 6 的结果，格式为 $\langle \text{key}=ID_i, \text{value}=\boldsymbol{S}_i\rangle$、$\langle \text{key}=id_i, \text{value}=\boldsymbol{S}_i^*\rangle$。

当 $i=1,2,\cdots,size(D_k)$、$j=1,2,\cdots,n$；$i=1,2,\cdots,size(Y_k)$、$j=1,2,\cdots,n$ 时，如果 $\boldsymbol{S}_i$、$\boldsymbol{S}_i^*$ 中的第 λ 位均分别大于 0，则将 $\boldsymbol{S}_i$、$\boldsymbol{S}_i^*$ 中的第 λ 位分别记为 1，否则为 0，$\boldsymbol{s}_k=[\boldsymbol{S}_i]$、$\boldsymbol{S}_k^*=[\boldsymbol{s}_k]$。

(2) 输出 $\langle \text{key}=ID, \text{value}=\boldsymbol{s}_k\rangle$、$\langle \text{key}=id, \text{value}=\boldsymbol{S}_K^*\rangle$ 并以检索特征数据库子向量 $\boldsymbol{d}_{k+2}=(ID,\boldsymbol{s}_k)$、检索特征子向量 $\boldsymbol{r}_k=(id,\boldsymbol{S}_k^*)$ 的形式存储于 HDFS 中，且 $\boldsymbol{D}_{k+2}=[\boldsymbol{d}_{k+2}]$、$\boldsymbol{R}_K=[\boldsymbol{r}_k]$。

8) 算法 8：Reduce 部分

(1) 输入算法 7 中 Map 的结果 $\boldsymbol{d}_{k+2}$、$\boldsymbol{r}_k$。

(2) 依据值 $\boldsymbol{S}_k$、$\boldsymbol{S}_k^*$ 进行归并计算。

(3) 输出 $\langle \text{key}=s_1,\cdots,s_k, \text{value}=(ID_1,\cdots,ID_k)\rangle$；$\langle \text{key}=\boldsymbol{S}_1^*,\cdots,\boldsymbol{S}_k^*, \text{value}=(id_1,\cdots,id_k)\rangle$。

在输出结果中，相似的检索向量 $\boldsymbol{S}_i$、$\boldsymbol{S}_i^*$ 分别被聚到一起，根据 ID、id 将其对应的相似检索子向量 $\boldsymbol{d}_{k+2}$、$\boldsymbol{r}_k$ 分别存储于 HDFS 中，且 $\boldsymbol{D}_{k+2}=[\boldsymbol{d}_{k+2}]$、$\boldsymbol{R}_K=[\boldsymbol{r}_k]$。

过程 4：构建相似度检索向量 $\boldsymbol{V}_k$。

输入：输入 HDFS 中 $\boldsymbol{D}_{k+2}$、$\boldsymbol{R}_K$ 以及相似度判定阈值 μ；

输出：相似度检索向量 $\boldsymbol{V}_k$。

9) 算法 9：Map 部分

(1) 输入 HDFS 中的中 $\boldsymbol{D}_{k+2}$、$\boldsymbol{R}_K$ 以及阈值 μ。

(2) 在 $\boldsymbol{D}_{k+2}$、$\boldsymbol{R}_K$ 的元素上，基于倒排索引算法对两者前 $r(1-\dfrac{\mu}{b})$ 位做异或运算，其中 r 是 $\boldsymbol{D}_{k+2}$、$\boldsymbol{R}_K$ 中元素的维度，μ 是相似度判定阈值，b 是首个元素二进制串维度。

(3) 若数据中 1 的个数小于 μ，则输出 $\langle \text{key}=ID_i, \text{value}=1\rangle$。

10) 算法 10：Reduce 部分

(1) 输入 Map 结果 $\langle \text{key}=ID_1, \text{value}=1\rangle$。

(2) 以 key 为 1 进行归并运算。

(3) 输出 $\langle \text{key}=1, \text{value}=(ID_1,ID_2,\cdots)\rangle$。

在输出的结果中，能被聚到一起的向量是具有相似性的向量，根据 ID 号将其对应的检索以相似度检索子向量 $\boldsymbol{v}_K$ 存储于 HDFS 中，且 $\boldsymbol{V}_k=[\boldsymbol{v}_K]$。

过程 5：计算效用值检索向量 $\boldsymbol{T}_k$。

输入：输入算法 10 的结果 $\boldsymbol{V}_k=[\boldsymbol{v}_K]$；

输出：效用值检索向量 $\boldsymbol{T}_k$。

11) 算法 11：Map 部分

(1) 输入 HDFS 中的相似度检索向量 $\boldsymbol{V}_k$，读取数据的格式为 $\langle \text{key}=ID_1, \text{value}=1\rangle$。

(2) 对 $\boldsymbol{V}_k$ 进行多属性决策效用值计算。

(3) 输出 $\langle \text{key}=ID_i, \text{value}=U_i\rangle$，其中 U_i 是第 i 个元组对应的效用值。

12) 算法 12：Reduce 部分

(1) 输入 Map 结果 $\langle \text{key}=ID_i, \text{value}=U_i\rangle$。

(2) 以 U_i 为 key 进行归并运算。

(3) 输出 $\langle \text{key}=U_i, \text{value}=(ID_1,ID_2,\cdots)\rangle$；

在输出的结果中，效用值检索向量 $\boldsymbol{T}_k$，$T_k=[(U_i,ID_i)]$ 被聚到一起，根据 ID 号依次对效用值进行排序，并将其存储于 HDFS 中。

过程 6：获取综合检索向量 $\boldsymbol{Z}_k$。

输入：输入 HDFS 中效用值检索向量 $\boldsymbol{T}_k$；

输出：综合检索向量 $\boldsymbol{Z}_k$。

13) 算法 13：Map 部分

(1) 输入 HDFS 中 $\boldsymbol{T}_k=[(U_i,ID_i)]$。

(2) 综合 SimHash 相似度和效用值公式(11)计算 $P_i=(1\leqslant i\leqslant k)$ 的综合评分 $P_{\text{roRank}\,i}$，其值为

$$P_{\text{roRank}\,i}=\lambda S_{\text{imHash}}(\boldsymbol{V}_k)+(1-\lambda)U_i,(0\leqslant\lambda\leqslant1) \quad (12)$$

由式(12)可知，当平衡因子 $\lambda=1$ 时，系统只进行基于改进 SimHash 的信息检索，并不会对检索到的输电监测数据进行效用值比较；当 $\lambda=0$ 时，则只进行输电监测数据的效应值比较，并不会考虑检索的输电监测数据与检索数据的 SimHash 匹配。

(3) 输出 $\langle\text{key}=ID_i,\text{value}=P_{\text{roRank}\,i}\rangle$。

14) 算法 14：Reduce 部分

(1) 输入 $\langle\text{key}=ID_i,\text{value}=P_{\text{roRank}\,i}\rangle$。

(2) 以 $P_{\text{roRank}\,i}$ 为 key 归并运算。

(3) 输出 $\langle\text{key}=P_{\text{roRank}\,i},\text{value}=(\cdots ID_i)\rangle$。

在输出的结果中，基于相似性程度对综合检索向量 $\boldsymbol{Z}_k$ 聚类，并根据 ID 号，以综合得分为指标进行排序存储于 HDFS 中，并输出排序第一的元组作为检索结果。在算法执行过程中，边检索边合并，以提高检索的效率。其中，过程 1、过程 2 是检索记录过程；过程 3、过程 4 是检索相似度计算；过程 5、过程 6 为检索相似度多属性决策及综合评分计算，MR-IRM 算法的时间复杂度为所有过程的复杂度之和，即 $O(|i|(3|n|+|j|+|r|))+O(|i|(|r|(1-\frac{3}{b})))+O(|i|(|k|+|i|))$。

5　算例分析

为验证 MR-IRM 算法的可行性，本文在 Hadoop 平台搭建集群实验环境，共 16 个节点：1 个管理、1 个 IO、14 个计算节点。系统采用 Rocky Linux 构建计算集群，配置 11th Gen Intel(R) Core(TM) i5-1135G7@2.40GHz 2.42 GHz，16 GB 内存，500 GB 硬盘，Hadoop-3.3.0。

5.1　算例数据描述

实验数据来源于电网某电力公司 2021 年第 3 季度输电线路在线监测的文本故障数据，包括电网实时运行监测数据和历史数据。结合气候(温度、湿度等)、日类型(休息日、节假日)、负荷，构成完整数据集。详细的数据集包含 2021 年 8 月 1 日 00:00 至 2021 年 9 月 9 日 24:00 的温度、湿度、负荷、86 个用户的样本数据以及日类型，初始数据格式如表 1 所示。

5.2　检索性能评价指标

判定输电监测数据检索性能的主要指标包括检索查询的查准率(PR)和查全率(RE)。查准率的计算公式为

表 1　电力数据初始格式

Table 1 Initial format of power data

日期	时间	最高温度/℃	最低温度/℃	相对湿度/%	周末	节假日	负荷/MW
2021-08-01	00:00	32	16	82	是	否	5.88
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2021-09-09	24:00	30	15	80	否	否	5.99

$$P_{\text{R}}=\frac{\sum\limits_{S_{\text{atisfy}}}v_{\text{ector}}}{\sum\limits_{A_{\text{ll}}}v_{\text{ector}}}\times100\% \quad (13)$$

式中：$\sum\limits_{A_{\text{ll}}}v_{\text{ector}}$ 是指在检索过程中检索出的所有元组的数量；$\sum\limits_{S_{\text{atisfy}}}v_{\text{ector}}$ 是 $\sum\limits_{A_{\text{ll}}}v_{\text{ector}}$ 中，符合检索需求的元组的数量。

查全率计算公式为

$$R_{\text{E}}=\frac{\sum\limits_{R_{\text{etrieve}}}v_{\text{ector}}}{\sum\limits_{S_{\text{ystem}}}v_{\text{ector}}}\times100\% \quad (14)$$

式中：$\sum\limits_{R_{\text{etrieve}}}v_{\text{ector}}$ 是指检索出的且与输电监测数据相关的元组的数量；$\sum\limits_{S_{\text{ystem}}}v_{\text{ector}}$ 是检索系统中所有元组的数量。

检索出的所有符合检索需求的输电监测数据，可根据式(12)预设一个参数 λ(平衡因子)进行计算，所得的查准率越高，表示算法性能越佳。明显地，输电监测数据查准率及检索结果的相关排序，会受 λ 取值影响。因此，为了选择式(12)中最优参数 λ，在分析实验结果之后，λ 与检索查准率的关系如图 2 所示，由图 2 可见，实验中 λ 取 0.34。

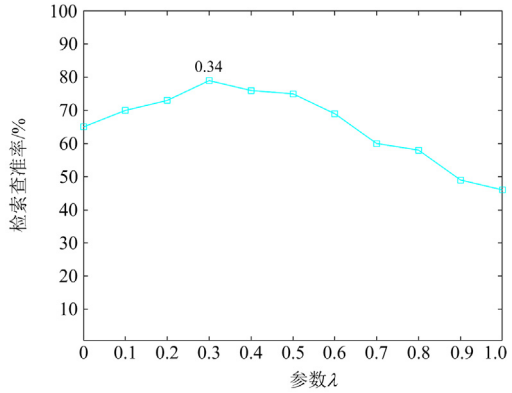

图 2　参数λ与检索查准率的关系

Fig. 2 Relationship between λ and retrieval precision

5.3 算法的查准率、查全率及效率分析

为评估 MR-IRM 的查准率、查全率及效率，实验中，将本文的检索结果与文献[23-26]所涉及到的 4 种检索算法的检索结果进行对比。其中：YD 为基于语义相似度和多属性决策商品智能检索算法；YZ 为基于元数据的知识组织智能检索算法；DQ 为多级索引驱动的地名信息检索方法；GS 为基于关键词的关系数据库时态信息检索方法；检索精度的评价标准用查准率(PR)、查全率(RE)及检索时间(TI)来衡量；检索迭代时间间隔为 0.2 s，迭代次数为 6，数据集为 200 kB(2500 条)。文献[23-26]中算法的查准率、查全率及检索时间如表 2 所示。

表 2 4 种算法的精确性比较

Table 2 Accuracy comparison of four algorithms

算法	数据源	P_R/%	R_E/%	T_I/s
YD	公司 1	79.15	86.12	5.32
YZ	公司 1	74.85	90.30	3.22
DQ	公司 1	66.42	57.97	11.69
GS	公司 1	42.36	19.24	6.82

由表 2 可见，YZ 算法的查全率与检索时间的平衡性最好，其性能最优；GS 算法的查准率、查全率最低且检索时间也相对较久、平衡性最差、检索性能最差。

同样地，采用小规模数据集验证 MR-IRM 的精度，且与基于 SimHash 检索的结果进行比较。实验设定了 6 个 Map 和 6 个 Reduce 进程；数据包含了 12 个属性，数据规模为 200 kB(2500 条)，算法的检测结果如表 3 所示。

表 3 MR-IRM 与 SimHash 算法检索结果

Table 3 Search results of MR-IRM and SimHash algorithm

算法	数据源	数据量/kB	P_R/%	P_E/%	T_I/s
SimHash	公司 1	200	38.13	85.41	5.24
MR-IRM	公司 1	200	95.78	91.88	3.31
YZ	公司 1	200	75.25	90.08	3.32

由表 3 可见：MR-IRM 算法具有最优的检索性能。在实验中，实验数据样本规模为 200 kB，首先利用基于 NLPIR PARSER 进行分词和关键词提取，构建关键词检索数据库，并引入改进的 TF-IDF 等模型，完成权重计算、加权和规范化，进一步构建序列检索数据库。同时，基于改进 SimHash 算法进行数据的合并及降维，构建检索特征数据库，并进一步基于海明距离计算模型构建相似度检索向量。其次，引入效用值模型(见式(9)—式(11))完成相似度计算，并基于 MapReduce 模型可与数学函数结合使用的特性，完成 Map 过程的分片和 Reduce 过程的归并，实现数据并行检索建模，从而实现 MR-IRM 算法并行计算。最后根据式(13)完成数据检索精度的计算。结果表明，MR-IRM 算法的 PR 值为 95.78%，取整后 PR 约为 96%。RE 值约为 92%，TI 值约为 3 s；此外，MR-IRM 算法的 PR 值比传统的 SimHash 算法约提高了 59.63%，RE 值约提高了 7.62%，TI 值约提高了 39.75%，即 MR-IRM 算法精确更高。

为进一步验证 MR-IRM 算法检索性能的健壮性，将 MR-IRM 与传统 SimHash 算法、表 2 中检索性能最优的 YZ 算法和检索性能最差 GS 算法进行对比。实验中，设定 12 个 Map 和 12 个 Reduce 进程，实验数据为 1.5 GB，每条记录包含 11 个属性，所得实验结果见图 3—图 5。

由图 3—图 5 可知：同等条件下，MR-IRM 和 YZ 算法的执行效率最高，但是 MR-IRM 算法的查准率及查全率均高于 YZ 算法，故总的来说，MR-IRM 算法性能最优，这是因其在执行过程中直接输出相似度检索向量，并对相似度检索向量进行多属性决策和综合评分，且多进程边合并边检索。此外，与 SimHash 相比，MR-IRM 算法检索效率约提高了 26.36%，其 PR、RE 和效率最高，故其检索性能也最优。

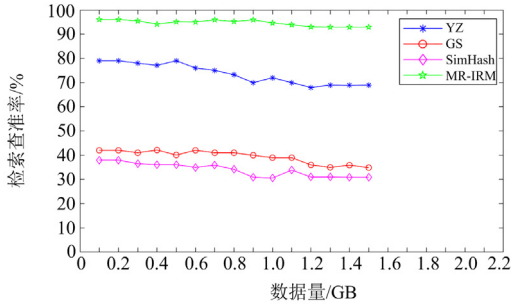

图 3 4 种算法检索查准率对比

Fig. 3 Comparison of retrieval precision of four algorithms

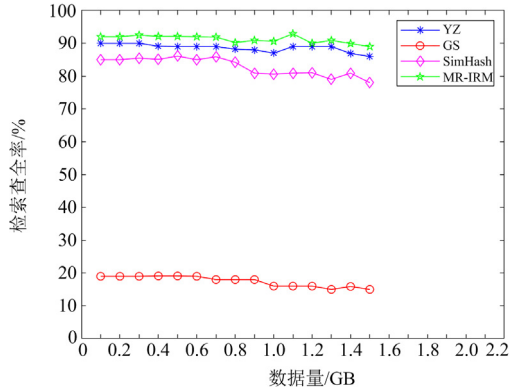

图 4 4 种算法检索查全率对比

Fig. 4 Comparison of retrieval recall rates of four algorithms

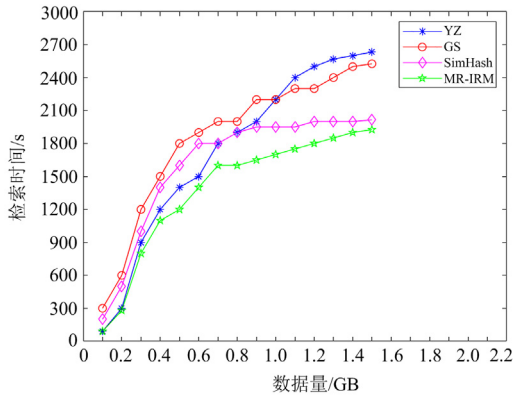

图 5 4 种算法检索运行时间对比

Fig. 5 Comparison of retrieval running time of four algorithms

5.4 参数对算法的影响

5.4.1 数据规模与 PR、RE 的关系

分别对 2.048 GB、20.48 GB、204.8 GB 的数据量进行 8 个 MapReduce 任务进程运行，来验证数据规模对 MR-IRM 和 SimHash 算法的 PR 和 RE 的影响，实验结果见表 4。

表 4 数据量对算法的影响

Table 4 Impact of data volume on the algorithm

算法	数据源	数据量/GB	P_R/%	R_E/%
SimHash	公司 1	2.048	36.47	80.55
		20.48	35.46	83.11
	公司 2	20.48	32.89	84.64
		204.8	29.76	86.73
MR-IRM	公司 1	2.048	95.71	91.36
		20.48	96.89	90.78
	公司 2	20.48	94.52	91.51
		204.8	95.37	92.48

由表 4 可知，数据量以 10 倍的规模增加时，对 MR-IRM 算法的 PR 和 RE 造成的误差均在 2%以内，SimHhash 的误差大于 2，且性能大大低于 MR-IRM 算法。因为不同的数据源其数据结构等存在差异，所以 MR-IRM 算法的检索性能会出现较小的浮动(PR、RE 值均在 2%以内浮动)，但足以满足海量输电监测数据相似度检索的实时性要求；然而，SimHash 算法检索所得 PR 值在 6.6% 以内浮动、RE 值在 2.5%以内浮动，因此在不同数据源中，数据规模对 MR-IRM 算法的影响更小、更稳定，更适用于迅速增长的海量输电监测数据的智能检索。

5.4.2 数据规模与检索时间的关系

实验设定 8 个 MapReduce 任务进程，所采用的实验数据包含 12 个属性，其实验数据分别为 10、30、60、100、300、500 MB，MR-IRM 检索时间与数据规模的关系见表 5。

由表 5 可知，本文实验数据环境下，MR-IRM 算法的检索效率受数据规模的影响较小。也就是说，随着数据量的增加，检索时间明显增加，但是随着数据量进一步增加时，检索时间上涨趋势变缓。这是因为数据规模较大时，可以充分发挥 MR-IRM 算法的并行性。MR-IRM 算法在所有并行任务启动时，仍然可以保持一定的检索效率，故本文提出的 MR-IRM 算法适用于海量电力数据智能检索。

表 5 检索时间与数据规模之间的关系

Table 5 Relationship between running time and data size

数据量/MB	检索时间/s
10	156.11
30	312.07
60	501.91
100	935.43
300	1202.86
500	1355.37

5.4.3 加速比和可伸缩性

节点数是系统加速比的衡量，代表着系统并行执行进程的数目。加速比(speedup)指同一个任务在并行处理器系统中运行消耗的时间的比率，用来衡量数据规模固定、不断增加节点数时并行算法的性能和效果，理想的加速比呈线性变化。可伸缩性(scalability)指以更大的规模来提高性能及可用性，表示节点数及数据规模都成比增加时并行算法的性能。对算法加速比的评估，实验采用 20 GB 的输电监测数据量，其规模分别为 2.5、5、10、20 GB；对算法可伸缩性的评定，节点数分别为 2、4、8、16，所得结果如图 6 所示。

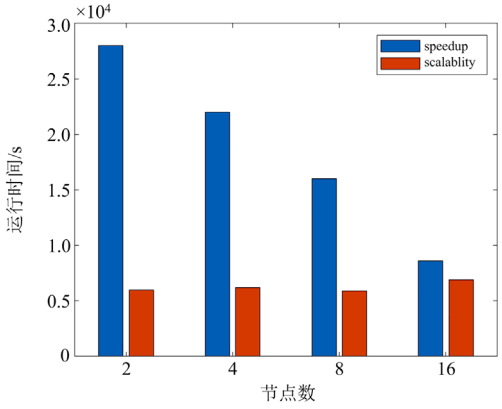

图 6 不同节点数的加速比与可伸缩性

Fig. 6 Speedup ratio and scalability of each number of nodes

由图 6 可知，本文所提 MR-IRM 算法具备较高的加速比及良好的可伸缩性，具有一定的适用性。

由于计算机通信开销等问题依然存在，故加速比无法达到理想的状态，这是因为受到计算机软、硬件资源消耗等的影响，当节点数为 16 时，算法的运行消耗时间小幅上涨，但算法的整体执行效率浮动范围较小，控制在 0.4%以内，总体具备良好的可伸缩性。

6 结论

1) MR-IRM 算法的查准率(PR)大于 95%、查全率(RE)大于 91%，且查准率、查全率及检索时间呈负相关性，算法高效、稳定，数据规模对其 PR、RE 值产生影响较小，具备良好的检索性能和较可靠的适用性。

2) MR-IRM 算法并行检索效率受数据规模影响较小，且多属性决策及综合评分进行相似性检索时，能智能推荐输出排名靠前的相似向量，算法的执行效率高、适用性强。

3) MR-IRM 算法具备极佳的加速比和可伸缩性，适用于海量电力运行数据的智能检索，并在行业故障检索系统的相关模块中取得了实践应用，效果显著。下一步将研究海量电力运行数据的寻优算法，例如更大规模节点数对算法检索性能产生的影响等，使其更好地运用于海量输电监测数据中，为企业提供更优的数据检索支持。

参考文献

[1] 王彩霞, 时智勇, 梁志峰, 等. 新能源为主体电力系统的需求侧资源利用关键技术及展望[J]. 电力系统自动化, 2021, 45(16): 37- 48.
WANG Caixia, SHI Zhiyong, LIANG Zhifeng, et al. Key technologies and prospects of demand-side resource utilization for power systems dominated by renewable energy[J]. Automation of Electric Power Systems, 2021, 45(16): 37-48.

[2] 丁斌, 袁博, 郑焕坤, 等. 基于大数据分析的电力信息系统安全状态监测技术研究[J]. 电测与仪表, 2021, 58(11): 59-66.
DING Bin, YUAN Bo, ZHENG Huankun, et al. Research on security state monitoring technology of power information system based on big data analysis[J]. Electrical Measurement & Instrumentation, 2021, 58(11): 59-66.

[3] 董小瑞, 孙伟, 樊群才, 等. 基于 KLDA-INFLO 的继电保护整定数据异常识别方法[J]. 电力科学与技术学报, 2022, 37(6): 132-137, 149.
DONG Xiaorui, SUN Wei, FAN Quncai, et al. A detection method for anomalies in protection relay setting based on the KLDA-INFLO[J]. Journal of Electric Power Science And Technology, 2022, 37(6): 132-137, 149.

[4] 梅玉杰, 李勇, 周王峰, 等. 基于机器学习的配电网异常缺失数据动态清洗方法[J]. 电力系统保护与控制, 2023, 51(7): 158-169.
MEI Yujie, LI Yong, ZHOU Wangfeng, et al. Dynamic data cleaning method of abnormal and missing data in a distribution network based on machine learning[J]. Power System Protection and Control, 2023, 51(7): 158-169.

[5] 郭琦, 卢远宏. 新型电力系统的建模仿真关键技术及展望[J]. 电力系统自动化, 2022, 46(10): 18- 32.
GUO Qi, LU Yuanhong. Key technologies and prospects of modeling and simulation of new power system[J]. Automation of Electric Power Systems, 2022, 46(10): 18-32.

[6] 宋雨露, 樊艳芳, 刘牧阳, 等. 基于 SC-DNN 和多源数据融合的新能源电力系统状态估计方法[J]. 电力系统保护与控制, 2023, 51(9): 177-187.
SONG Yulu, FAN Yanfang, LIU Muyang, et al. State estimation method of a new energy power system based on SC-DNN and multi-source data fusion[J]. Power System Protection and Control, 2023, 51(9): 177-187.

[7] 蔡银琼, 范意兴, 郭嘉丰, 等. 基于多表达的第一阶段语义检索模型[J]. 计算机工程与应用, 2023, 59(4): 139-146.
CAI Yinqiong, FAN Yixin, GUO Jiafeng, et al. Multi-representation model for the first-stage semantic retrieval[J]. Computer Engineering and Applications, 2023, 59(4): 139-146.

[8] 刘东, 张越, 皮俊波, 等. 面向电网调控信息智能检索的知识图谱构建及应用[J]. 中国电力, 2023, 56(7): 78-84.
LIU Dong, ZHANG Yue, PI Junbo, et al. Construction and application of knowledge graph for intelligent retrieval of power grid dispatching and control information[J]. Electric Power, 2023, 56(7): 78-84.

[9] GHAREHCHOPOGH F S, NAMAZI M, EBRAHIMI L, et al. Advances in sparrow search algorithm: a comprehensive survey[J]. Archives of Computational Methods in Engineering, 2023, 30(1): 427-455.

[10] ELGHAISH F, CHAUHAN J K, MATARNEH S, et al. Artificial intelligence-based voice assistant for BIM data management[J]. Automation in Construction, 2022, 140: 104320.

[11] HOFSTÄTTER S, CHEN J, RAMAN K, et al. Fid-light: efficient and effective retrieval-augmented text generation[C] // Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, July23-27, 2023, Taipei, China: 1437-1447.

[12] Gao L, Ma X, Lin J, et al. Tevatron: An efficient and

flexible toolkit for dense retrieval[J]. arxiv preprint arxiv, 2022(9): 22-34.

[13] SHYLA S I, SUJATHA S S. Efficient secure data retrieval on cloud using multi-stage authentication and optimized blowfish algorithm[J]. Journal of Ambient Intelligence and Humanized Computing, 2022, 2(6): 1-11.

[14] SHEYNIN S, ASHUAL O, POLYAK A, et al. Knn-diffusion: Image generation via large-scale retrieval[J]. arxiv preprint arxiv, 2022(9): 5-15.

[15] 周慧聪, 郭肇强, 梅元清, 等. 版本失配和数据泄露对基于缺陷报告的缺陷定位模型的影响[J]. 软件学报, 2023, 34(5): 2196-2217.
ZHOU Huicong, GUO Zhaoqiang, MEI Yuanqing, et al. Watch out for version mismatching and data leakage! a case study of their influence in bug report based bug localization models[J]. Journal of Software, 2023, 34(5): 2196-2217.

[16] 宋人杰, 余通, 陈宇红, 等. 基于MapReduce模型的大数据相似重复记录检测算法[J]. 上海交通大学报, 2018, 52(2): 214-221.
SONG Renjie, YU Tong, CHENG Yuhong, et al. A similar duplicate record detection algorithm for big data based on MapReduce[J]. Journal of Shanghai Jiaotong University, 2018, 52(2): 214-221.

[17] OGHBAIE M, ZANJIREH M M. Pairwise document similarity measure based on present term set[J]. Journal of Big Data, 2018, 5(1): 217-228.

[18] 叶雪梅, 毛雪岷, 夏锦春, 等. 文本分类 TF-IDF 算法的改进研究[J]. 计算机工程与应用, 2019, 55(2): 104-109, 161.
YE Xuemei, MAO Xuemin, XIA Jinchun, et al. Improved approach to TF-IDF algorithm in text classification[J]. Computer Engineering and Applications, 2019, 55(2): 104-109, 161.

[19] SUN Jing, GAN Xingjia, GONG Dunwei, et al. A self-evolving fuzzy system online prediction-based dynamic multi-objective evolutionary algorithm[J]. Information Sciences, 2022(20): 61-72.

[20] 叶子诚, 闫桂英. 基于图模型的关键词提取算法研究[J]. 系统科学与数学, 2021, 41(4): 967-975.
YE Zicheng, YA Guiying. Study on keyword extraction algorithm based on graphical model[J]. Journal of Systems Science and Mathematical Sciences, 2021, 41(4): 967-975.

[21] BALABANOVI M, SHOHAM Y. content-based collaborative recommendation[J]. Communication of the ACM, 2019, 40(3): 66-72.

[22] 葛晓琳, 徐轶胜, 符杨, 等. 考虑多重不确定性影响的海上风储多属性联合规划[J]. 电网技术, 2023, 47(10): 4140-4153.
GE Xiaolin, XU Yisheng, FU Yang, et al. Multi-attribute joint planning of offshore wind and storage considering influence of multiple uncertainties[J]. Power System Technology, 2023, 47(10): 4140-4153.

[23] 丁邱, 迟海洋, 严馨, 等. 基于 Transformer 模型的问句语义相似度计算[J]. 计算机工程与设计, 2023, 44(3): 887-893.
DING Qiu, CHI Haiyang, YAN Xin, et al. Semantic similarity calculation of questions based on Transformer model[J]. Computer Engineering and Design, 2023, 44(3): 887-893.

[24] 冯禹洪, 吴坤汉, 黄志鸿, 等. 基于 FP-tree 和 MapReduce 的集合相似度自连接算法[J]. 计算机研究与发展, 2023, 60(12):1-17.
FENG Yuhong, WU Kunhan, HUANG Zhihong, et al. A set similarity join algorithm with FP-tree and map reduce[J]. Journal of Computer Research and Development, 2023, 60(12):1-17.

[25] 王丹, 张祥合, 赵浩宇. 基于元数据的信息知识组织智能检索系统设计[J]. 情报科学, 2019, 37(9): 113-116.
WANG Dan, ZHANG Xianghe, ZHAO Haoyu. Design of information and knowledge organization intelligent retrieval system based on metadata[J]. Information Science, 2019, 37(9): 113-116.

[26] 李佩, 陈乔松, 陈鹏昌, 等. 基于模态特异及模态共享特征信息的多模态细粒度检索[J]. 计算机工程, 2022, 48(11): 1000-3428.
LI Pei, CHENG Qiaosong, CHEN Pengchang, et al. Multimodal fine-grained retrieval based on modality-specific and modality-shared feature information[J]. Computer Engineering, 2022, 48(11): 1000-3428.

收稿日期：2023-06-01；　修回日期：2023-08-09

作者简介：

赵松燕(1994—)，女，通信作者，硕士研究生，研究方向为模式识别、信息处理、电力数据智能处理及应用、工业网络安全等；E-mail: 1032322086@qq.com

曲朝阳(1963—)，男，博士，教授，博士生导师，研究方向为电力大数据处理及应用。E-mail: 862888620@qq.com

（编辑　姜新丽）