Введение в информационный поиск

Лабораторная работа №2

Списки словопозиций

Выполнил: студент группы 2371 Тарасевич Екатерина Владимировна

Преподаватель: Зинченко А.С.

6 ноября 2020 г.

1 Вопросы

- Сколько всего лексем содержится в коллекции (значение атрибута countTokens)?
- Чему равен размер словаря (размер атрибута index)?
- Этап 1
    1. Термов в коллекции - 888378
    2. Размер словаря - 25964
- Этап 2
    1. Термов в коллекции - 467011
    2. Размер словаря - 25433
- Этап 3
    1. Термов в коллекции - 480245
    2. Размер словаря - 17420

2 Листинг программы

Листинг 1. Класс Inverted Index

```
1  public class InvertedIndex {
2      List<String> documents = new ArrayList<String>();
3      Map<String, Pair> index = new HashMap<String, Pair>();
4      int docID = 0;
5      int countTokens = 0;
6      ArrayList<String> stopWords = new ArrayList<>();
7
8      public InvertedIndex(String path) throws IOException {
9          File file = new File(path);
10         Scanner sc = new Scanner(file);
11
12         while (sc.hasNextLine()) {
13             stopWords.add(sc.nextLine().toLowerCase());
14         }
15     }
16
17
18     public void indexDocument(String path) throws IOException {
19
20         File file = new File(path);
21         System.out.println(file.getName());
22         Document doc = Jsoup.parse(file, "UTF-8");
23
24
25         if (!this.documents.contains(file.getName())) {
26             this.documents.add(file.getName());
27
28
29             int docID = this.documents.size() - 1;
30             Elements body = doc.select("body");
31             String line;
32             Pair idx;
33             String s = doc.body().text();
34             Scanner in = new Scanner(s);
35
36             while (in.hasNextLine()) {
37                 line = in.nextLine();
38                 line = line.toLowerCase();
39
```

```
40                 for (String word : line.split("[^a-zA-Z0-9_']+")) {
41                     if (!stopWords.contains(word)) {
42 
43                         Stemmer stemmer = new Stemmer();
44 
45                         stemmer.add(word.toCharArray(), word.length());
46                         stemmer.stem();
47                         String term = stemmer.toString();
48 
49                         if (index.containsKey(term)) {
50                             index.get(term).addDocument(docID);
51                         } else {
52                             idx = new Pair(docID);
53                             index.put(term, idx);
54                         }
55                         countTokens++;
56                     }
57                 }
58             }
59             System.out.println("docID: " + docID);
60             System.out.println("path: " + file.getPath());
61             System.out.println("index.size(): " + this.index.size());
62             System.out.println("------------------------------------");
63             System.out.println(countTokens);
64             System.out.println("----------------------------");
65         }
66 
67 
68     }
69 
70 
71     public void indexCollection(String folder) throws IOException {
72         File dir = new File(folder);
73         String[] files = dir.list();
74         for (int i = 0; i != files.length; i++) {
75             this.indexDocument("collection_html\\" + files[i]);
76         }
77     }
78 
79     public List<Integer> getIntersection(List<Integer> list1, List<Integer>
    list2) {
80         List<Integer> answer = new ArrayList<Integer>();
```

```
81          int a;
82          int b;
83          ListIterator<Integer> iterator1 = list1.listIterator();
84          ListIterator<Integer> iterator2 = list2.listIterator();
85
86          while (iterator1.hasNext() && iterator2.hasNext()) {
87              a = iterator1.next();
88              b = iterator2.next();
89              if (a == b) {
90                  answer.add(a);
91              } else if (a < b) {
92                  iterator2.previous();
93              } else iterator1.previous();
94          }
95          return answer;
96      }
97
98
99      public List<Integer> executeQuery(String query) {
100         String[] s = query.toLowerCase().split(" and ");
101         List<String> words = new ArrayList<>();
102
103         Stemmer stemmer = new Stemmer();
104
105         for (int i = 0; i < s.length; i++) {
106
107             String word = s[i];
108             stemmer.add(word.toCharArray(), word.length());
109             stemmer.stem();
110             String term = stemmer.toString();
111
112
113             if (!stopWords.contains(s[i])) {
114                 words.add(term);
115             }
116         }
117
118         List<Integer> res = new LinkedList<>();
119         if (words.size() > 0 && index.get(words.get(0)) != null) {
120             res = index.get(words.get(0)).list;
121         }
122
```

```
123         for (int i = 1; i < words.size(); i++) {
124
125             if (res != null && index.get(words.get(i)) != null) {
126                 res = getIntersection(res, index.get(words.get(i)).list);
127             } else {
128                 res = new LinkedList();
129                 break;
130             }
131         }
132         return res;
133     }
134 }
```

Листинг 2. Класс Pair

```
1  class Pair {
2      int termFrequency;
3      LinkedList<Integer> list;
4
5
6      Pair(int docID) {
7          this.termFrequency = 1;
8          this.list = new LinkedList<>();
9          this.list.add(docID);
10
11     }
12
13     void addDocument(int docID) {
14
15         if ((this.list.isEmpty() || docID > this.list.getLast())) {
16             list.add(docID);
17             this.termFrequency++;
18         }
19
20     }
21 }
```

Листинг 3. Класс Main

3 Результаты работы программы

- Результаты по запросу: bless
  1. ID 0 All's Well That Ends Well Entire Play.htm
  2. ID 1 Antony and Cleopatra Entire Play.htm
  3. ID 2 As You Like It Entire Play.htm
  4. ID 3 Comedy of Errors Entire Play.htm
  5. ID 4 Coriolanus Entire Play.htm
  6. ID 5 Cymbeline Entire Play.htm
  7. ID 6 Hamlet Entire Play.htm
  8. ID 7 Henry IV, part 1 Entire Play.htm
  9. ID 8 Henry IV, part 2 Entire Play.htm
  10. ID 9 Henry V Entire Play.htm
  11. ID 10 Henry VI, part 1 Entire Play.htm
  12. ID 11 Henry VI, part 2 Entire Play.htm
  13. ID 12 Henry VI, part 3 Entire Play.htm
  14. ID 13 Henry VIII Entire Play.htm
  15. ID 15 King John Entire Play.htm
  16. ID 16 King Lear Entire Play.htm
  17. ID 17 Love's Labour's Lost Entire Play.htm
  18. ID 18 Macbeth Entire Play.htm
  19. ID 19 Measure for Measure Entire Play.htm
  20. ID 20 Merchant of Venice Entire Play.htm
  21. ID 21 Merry Wives of Windsor Entire Play.htm
  22. ID 22 Midsummer Night's Dream Entire Play.htm
  23. ID 23 Much Ado About Nothing Entire Play.htm
  24. ID 24 Othello Entire Play.htm
  25. ID 25 Pericles Entire Play.htm
  26. ID 26 Richard II Entire Play.htm
  27. ID 27 Richard III Entire Play.htm
  28. ID 28 Romeo and Juliet Entire Play.htm
  29. ID 29 Taming of the Shrew Entire Play.htm
  30. ID 30 The Tempest Entire Play.htm
  31. ID 31 Timon of Athens Entire Play.htm
  32. ID 32 Titus Andronicus Entire Play.htm
  33. ID 33 Troiles and Cressida Entire Play.htm
  34. ID 34 Twelfth Night Entire Play.htm

35. ID 35 Two Gentlemen of Verona Entire Play.htm
36. ID 36 Winter's Tale Entire Play.htm

- Результаты по запросу: Brutus

1. ID 1 Antony and Cleopatra Entire Play.htm
2. ID 4 Coriolanus Entire Play.htm
3. ID 6 Hamlet Entire Play.htm
4. ID 9 Henry V Entire Play.htm
5. ID 14 Julius Caeser Entire Play.htm
6. ID 32 Titus Andronicus Entire Play.htm

- Результаты по запросу: by

1. Ничего не найдено :(

- Результаты по запросу: blessing

1. ID 0 All's Well That Ends Well Entire Play.htm
2. ID 1 Antony and Cleopatra Entire Play.htm
3. ID 2 As You Like It Entire Play.htm
4. ID 3 Comedy of Errors Entire Play.htm
5. ID 4 Coriolanus Entire Play.htm
6. ID 5 Cymbeline Entire Play.htm
7. ID 6 Hamlet Entire Play.htm
8. ID 7 Henry IV, part 1 Entire Play.htm
9. ID 8 Henry IV, part 2 Entire Play.htm
10. ID 9 Henry V Entire Play.htm
11. ID 10 Henry VI, part 1 Entire Play.htm
12. ID 11 Henry VI, part 2 Entire Play.htm
13. ID 12 Henry VI, part 3 Entire Play.htm
14. ID 13 Henry VIII Entire Play.htm
15. ID 15 King John Entire Play.htm
16. ID 16 King Lear Entire Play.htm
17. ID 17 Love's Labour's Lost Entire Play.htm
18. ID 18 Macbeth Entire Play.htm
19. ID 19 Measure for Measure Entire Play.htm
20. ID 20 Merchant of Venice Entire Play.htm
21. ID 21 Merry Wives of Windsor Entire Play.htm
22. ID 22 Midsummer Night's Dream Entire Play.htm
23. ID 23 Much Ado About Nothing Entire Play.htm

24. ID 24 Othello Entire Play.htm
25. ID 25 Pericles Entire Play.htm
26. ID 26 Richard II Entire Play.htm
27. ID 27 Richard III Entire Play.htm
28. ID 28 Romeo and Juliet Entire Play.htm
29. ID 29 Taming of the Shrew Entire Play.htm
30. ID 30 The Tempest Entire Play.htm
31. ID 31 Timon of Athens Entire Play.htm
32. ID 32 Titus Andronicus Entire Play.htm
33. ID 33 Troiles and Cressida Entire Play.htm
34. ID 34 Twelfth Night Entire Play.htm
35. ID 35 Two Gentlemen of Verona Entire Play.htm
36. ID 36 Winter's Tale Entire Play.htm

- Результаты по запросу: archer

  1. ID 10 Henry VI, part 1 Entire Play.htm
  2. ID 23 Much Ado About Nothing Entire Play.htm
  3. ID 25 Pericles Entire Play.htm
  4. ID 27 Richard III Entire Play.htm
  5. ID 32 Titus Andronicus Entire Play.htm

- Результаты по запросу: Brutus AND Caesar AND Calpurnia

  1. ID 14 Julius Caeser Entire Play.htm

- Результаты по запросу: Brutus AND Caesar AND Calpurnia AND before

  1. ID 14 Julius Caeser Entire Play.htm

- Результаты по запросу: SpiderMan AND Brutus AND Caesar

  1. Ничего не найдено :(

- Результаты по запросу: villain AND hero

  1. ID 0 All's Well That Ends Well Entire Play.htm
  2. ID 2 As You Like It Entire Play.htm
  3. ID 6 Hamlet Entire Play.htm
  4. ID 23 Much Ado About Nothing Entire Play.htm
  5. ID 28 Romeo and Juliet Entire Play.htm