

cje3.py

```
1 # cos類似度から、類似文書を表示する。知識情報演習3で作成。
2
3
4 from IPython.terminal.prompts import Token
5 import math
6 import pandas as pd
7 import re
8 from janome.tokenizer import Tokenizer
9
10 t = Tokenizer()
11
12 index_file = "index/index_nostop.txt"
13 idf_scores = {}
14 tfidf_scores = {}
15 query_tf = {}
16 query_tfidf = {}
17 stopwords = {}
18 query_words = {}
19 ranking_docs = []
20
21 f = open(index_file, 'r')
22
23 # indexから文書のデータフレームを作成 -----
24 for line in f:
25     line = line.rstrip()
26     split_line = line.split('\t')
27     word = split_line[0]
28     doc = split_line[1]
29     idf = float(split_line[2])
30     tfidf = float(split_line[3])
31
32     if word not in tfidf_scores:
33         tfidf_scores[word] = {}
34         tfidf_scores[word][doc] = tfidf
35     else:
36         tfidf_scores[word][doc] = tfidf
37
38     if word not in idf_scores:
39         idf_scores[word] = {}
40         idf_scores[word] = idf
41     else:
42         idf_scores[word] = idf
43
44 tfidf_table = pd.DataFrame(tfdif_scores)
45 tfidf_table = tfidf_table.fillna(0)
46
47
48 # クエリのデータフレームを作成 -----
49 query = '日本 ワールドカップ'
50 #query = 'ナゲツ ファイナル'
51 #query = 'NBA トレード'
52 #query = '渡邊雄太 NBA'
53 #query = 'ウェンバンヤマ ドラフト'
54 query_file = 'query'
55
56 argv = query.split()
```

```
57 argc = len(argv)
58 # クエリ表示
59 print(argv)
60
61 pattern = re.compile(r"^[ --]$")
62 stopwords['という'] = 1
63 stopwords['にて'] = 1
64
65 tokens = []
66 for i in range(argc):
67     list = t.tokenize(argv[i])
68     tokens.append(list)
69 flat_t = [item for sublist in tokens for item in sublist]
70
71 flat_token = []
72 for token in flat_t:
73     tmp = token.surface
74     judge = pattern.match(tmp)
75     flat_token.append(tmp)
76
77 if judge:
78     continue
79 if tmp in stopwords:
80     continue
81
82 if tmp in query_words:
83     query_words[tmp] += 1
84 else:
85     query_words[tmp] = 1
86
87
88 for index_word in idf_scores:
89     query_tf[index_word] = {}
90     query_tfidf[index_word] = {}
91     query_tf[index_word][query_file] = 0
92     query_tfidf[index_word][query_file] = 0
93
94 for query_word in query_words:
95     if query_word in query_tf:
96         query_tf[query_word][query_file] += 1
97
98 for query_word in query_words:
99     if query_word in query_tf:
100         query_tfidf[query_word][query_file] = float(query_tf[query_word][query_file]) * float(idf_scores[query_word])
101
102 query_table = pd.DataFrame(query_tfidf)
103
104
105 # 各データフレームからcos類似度を計算 -----
106 for query_word in query_words:
107     if query_word in tfidf_scores:
108         for doc in tfidf_scores[query_word]:
109             ranking_docs[doc] = 1
110
111 for doc in ranking_docs:
112     numerator = 0
113     doc_vec = tfidf_table.loc[doc]
114     query_vec = query_table.loc[query_file]
115     for i in range(len(query_vec.values)):
```

```
116     i_value = query_vec.values[i] * doc_vec.values[i]
117     numerator = numerator + i_value
118
119     denominator = 0
120     query_value = 0
121     doc_value = 0
122     for i in range(len(query_vec.values)):
123         query_value += query_vec.values[i]**2
124         doc_value += doc_vec.values[i]**2
125
126     denominator = math.sqrt(query_value) * math.sqrt(doc_value)
127     cosine = numerator / denominator
128
129     ranking_docs[doc] = cosine
130
131 for i in sorted(ranking_docs.items(), key=lambda x:x[1], reverse=True):
132     print('cos類似度:',i)
```