Information Processing and Retrieval Project Report – Part 2

Instituto Superior Técnico – Universidade de Lisboa

Margarida Costa  
 83425

Marta Aparício  
 83525

Paulo Alves  
 83538

ABSTRACT

Introduction

1 An approach based on graph ranking

1.1 Implementation

**1.2** Results

2 Improving the graph-ranking method

2.1 Implementation

2.2 Results

3 An unsupervised rank aggregation approach

3.1 Implementation

Combination methods tested were Reciprocal Rank Fusion (RRF), CombSum and CombMNZ. The features (ranking scores) considered were: term frequency (TF), inverse document frequency (IDF), TF-IDF and BM25 and graph centrality scores for the candidate. Graph centrality scores were computed through the library network.

3.2 Results

There is no better results setting thershlold in ignoring terms. No thresholds for CombMNZ. Centrality with co-occurrences.

Table 1 Mean Average Precison

|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **RRF** | **CombSum** | **CombMNZ** |
| tf | 0.03356 | 0.02953 | 0.02953 |
| idf | 0.01678 | 0.03740 | 0.03740 |
| tfidf | 0.02740 | 0.02526 | 0.02526 |
| bm25 | 0.03356 | 0.07480 | 0.07480 |
| bm25, tf | 0.02909 | 0.02229 | 0.02246 |
| bm25, idf, tf, tfidf | 0.01846 | 0.01432 | 0.02139 |
| bm25, centrality | 0.04120 | 0.02479 | 0.02622 |
| idf, centrality | 0.03191 | 0.01996 | 0.02615 |
| tf, centrality | 0.03805 | 0.01976 | 0.02506 |
| bm25, idf, centrality | 0.03088 | 0.01800 | 0.02528 |
| bm25, tf, centrality | 0.03406 | 0.01814 | 0.02461 |
| bm25, tfidf, centrality | 0.03269 | 0.01821 | 0.024374 |
| bm25, idf, tf, tfidf, centrality | 0.02561 | 0.01687 | 0.023357 |

Thresholds were set in CombMNZ.

Table 2 Mean Average Precison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Threshold** | **0.0001** | **0.001** | **0.01** | **0.1** |
| bm25, tf | 0.02246 | 0.023154 | 0.03048 | 0.01748 |
| bm25, idf, tf, tfidf | 0.02190 | 0.02190 | 0.03620 | 0.01963 |
| bm25,tf, tfidf | 0.02250 | 0.02250 | 0.03642 | 0.01996 |
| bm25, idf, tfidf | 0.02319 | 0.02319 | 0.03486 | 0.01910 |
| idf, tf | 0.02290 | 0.02290 | 0.03433 | 0.01957 |
| bm25, tfidf | 0.02129 | 0.02129 | 0.03434 | 0.01748 |
| idf, tf | 0.02290 | 0.02290 | 0.03433 | 0.01957 |
| idf, tfidf | 0.02292 | 0.022922 | 0.03307 | 0.01996 |
| 'tf', 'tfidf') | 0.02187 | 0.021875 | 0.03101 | 0.01921 |

It presents better results for the threshold 0.01.

Using scale factor and a threshold of 0.01 results inscreased about 0.010 on MAP.

Table 3 Mean Average Precision

|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **RRF** | **CombSum** | **CombMNZ** |
| bm25 | 0.06235 | 0.36801 | 0.21414 |
| bm25, tf | 0.11434 | 0.08777 | 0.06701 |
| bm25, idf, tf, tfidf | 0.13961 | 0.04610 | 0.05912 |
| bm25, tf, tfidf | 0.13260 | 0.05319 | 0.05223 |
|  |  |  |  |
|  |  |  |  |

4  A practical application

4.1 Implementation

For this part the goal was to implement a program that illustrated the keyphrase extraction. To do so, the libraries “wordcloud” and “matplotlib.pyplot” were used. To better show the keyphrases rank, the ranking charts were chosen. To be more concrete, the charts used were: Bar chart, Wordcloud char and lastly Spyder chart.

The charts present the scores given by the RRF, CombSum and CombMNZ combination methods, calculated by the functions of the previous exercise and using the tf, tfidf, idf and bm25 rankers.

4.2 Results

To better observe the results given by this exercise, open the .html generated by the exercise.