­­DMT – Homework1

*Vigèr Durand Azimedem Tsafack(1792126)*

*Malick Alexandre N. Sarr (1788832)*

* **First Step**: Search-Engine Evaluation

1. **Assessment of 3 search engines using: P@K, R-Precision, MRR and nDCG**

**(Python code:** *main\_part\_1\_1.py***, custom module:** *hwmodule.py***)**

*Statistics on the Dataset:*

|  |  |  |
| --- | --- | --- |
| Dataset | Number of Documents | Number of Queries |
| GT | 728 | 222 |
| SE\_1 | 1099 | 222 |
| SE\_2 | 1398 | 222 |
| SE\_3 | 1395 | 222 |

*P@k table:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Search Engine | Mean(P@1) | Mean(P@3) | Mean(P@5) | Mean(P@10) |
| SE\_1 | 0.031532 | 0.03003 | 0.027928 | 0.025676 |
| SE\_2 | 0.301802 | 0.295796 | 0.263063 | 0.185586 |
| SE\_3 | 0.238739 | 0.205706 | 0.186486 | 0.143243 |

*R-Precision table:*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Search Engine | Mean  (R-P\_Distr.) | Min  (R-P\_Distr.) | 1°\_quart.  (R-P\_Distr.) | Median  (R-P\_Distr.) | 3°\_quart.  (R-P\_Distr.) | MAX  (R-P\_Distr.) |
| SE\_1 | 0.022563 | 0 | 0 | 0 | 0 | 0.666667 |
| SE\_2 | 0.254943 | 0 | 0 | 0.25 | 0.428571 | 1 |
| SE\_3 | 0.179328 | 0 | 0 | 0.142857 | 0.333333 | 1 |

*MRR table:*

|  |  |
| --- | --- |
| Search Engine | MRR |
| SE\_1 | 0.081414 |
| SE\_2 | 0.486303 |
| SE\_3 | 0.395118 |

*nDCG table:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Search Engine | Mean  (nDCG@1) | Mean  (nDCG@3) | Mean  (nDCG@5) | Mean  (nDCG@10) |
| SE\_1 | 0.025521 | 0.061579 | 0.085546 | 0.126786 |
| SE\_2 | 0.195554 | 0.526353 | 0.672991 | 0.80411 |
| SE\_3 | 0.158024 | 0.388966 | 0.487971 | 0.613001 |

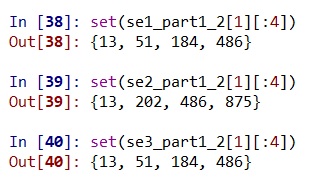
1. **Helping the “NoobDataScience” Company to choose between 3 different SE’s**

**(Python code:** *main\_part\_1\_2.py***, custom module:** *hwmodule.py***)**

The data we have here show that we are dealing with ranking algorithms. Each query retrieves 200 documents ranked in order of relevance. This simple analysis makes us understand that Precision, Recall and even their harmonic mean (F-measure) are not good metrics for picking the best search engine here.

However, since the “NoobDataScience” app will return to the user only four of the 200 query results, we will go forward assuming that the app picks the first four results returned by the ranking algorithm and show them to the user in a casual order.

The previous assumption changes everything. In fact, considering only the first query, the three search engines will return the following results:



In this particular case (case where the app returns four unranked results), we are dealing with a “normal” retrieval system without ranking. Thus, measures as P@K, MAP, MRR, ors are not useful since they take into account the rank.

The metric suiting the most our purpose here is the *mean(F-measure)*. We are not simply using *mean(P)* cause we want to take into account not only the precision but also the recall.

*F-measure table:*

|  |  |
| --- | --- |
| Search Engine | Mean  (F\_measure) |
| SE\_1 | 0.247949 |
| SE\_2 | 0.177724 |
| SE\_3 | 0.253298 |

This F-measure’s table tell us that the “NoobDataScience” company should choose the SE 3 since it has the greatest F-measure.

* **Second Step**: Near-Duplicates-Detection

1. **Finding all near-duplicate documents inside a dataset**

**(Python code:** *main\_part\_1\_2.py***, custom module:** *hwmodule.py***)**

dsadsasss