**Information retrieval final project**

This project was written by:

Team: BP

Alexandra Kabalik I.D. 322159450 email: chilikoa@post.bgu.ac.il

[Gill Michalovich](mailto:Gill%20Michalovich) I.D..315041426 email: gilmich@post.bgu.ac.il

Our link to the Github repo:

<https://github.com/AlexaKabalik/Final-project-IR>

A link to Google Storage Bucket:

<https://console.cloud.google.com/storage/browser?organizationId=536124907474&project=>alexandrakabalik&prefix=&forceOnObjectsSortingFiltering=false

List of all indexes files

All the pkl and bin files of inverted index creation

We start our model building from building a mini inverted indexes. This was done on a small piece of Wikipedia data (1000 documents) in the same way as it was in homework assignment 3. In the next step, we continue by building inverted indexes for entire Wikipedia data. This was done in the same ways as it was in the assignment 3. We direct all created files to our bucket (see link above). After, we have Wikipedia data for entire data we run the following test to optimize our model. First, we divide our new query training data set (that was provided by staff course) into 5 group of 6 queries each. Then we run the tests only on 4 groups, in each test we test the average time to retrieve and average MAP@40. At the final step, to test our model we test him on the lest group.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group1** | **Group2** | **Group3** | **Group4** | **Group5** |
| “best marvel movie” | “How to make wine at home?” | “The Simpsons” | “How do you make gold” | “World Cup 2022” |
| “How do kids come to world? | “Most expensive city in the world” | “World cup” | “Marijuana” | “Dolly the sheep” |
| “Information retrieval” | “India” | “How to lose weight” | “How to make hummus” | “Ciggarets” |
| “LinkedIn” | “how to make money fast? | “Java” | “Winter” | What is the best place to livein? |
| How to make coffee?” | “Netflix” | “Air Jordan” | “Rick and Morty” | Elon musk |
| “Ritalin” | “Apple computer” | “how to deal with depression?” | “Natural Language processing” | How do you breed flowers? |

We run the following test to evaluate our model as following: first we test our model by retrieving documents only by search\_body function and search\_title Exp1 and Exp2 respectively. Then, we test the contribution of each factor to our model by adding to each factor weight (Exp3-Exp10). The results of this experiments are summarized in the following tables.:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Exp 1** | | **Exp2** | | **Exp 3** | | **Exp 4** | | **Exp 5** | |
| **body\_w** | **1** | |  | | **0.5** | | **0.6** | | **0.7** | |
| **title\_w** |  | | **1** | | **0.5** | | **0.4** | | **0.5** | |
|  | **Time**  **avg** | **MAP@40**  **avg** | **Time**  **avg** | **MAP@40**  **avg** | **Time**  **avg** | **MAP@40**  **avg** | **Time**  **avg** | **MAP@40**  **avg** | **Time**  **avg** | **MAP@40**  **avg** |
| **Group1** | 2.45 | 0.399 | 0.106 | 0.488 | 3.70 | 0.387 | 3.81 | 0.384 | 3.69 | 0.384 |
| **Group2** | 3.45 | 0.456 | 0.107 | 0.286 | 5.26 | 0.432 | 5.38 | 0.414 | 5.26 | 0.42 |
| **Group3** | 1.67 | 0.424 | 0.103 | 0.275 | 3.17 | 0.508 | 3.22 | 0.499 | 3.22 | 0.508 |
| **Group4** | 1.51 | 0.463 | 0.05 | 0.627 | 1.94 | 0.289 | 1.96 | 0.301 | 1.94 | 0.289 |
| **Time avg** | 2.27 |  | 0.09 |  | 3.52 |  | 3.59 |  | 3.53 |  |
| **MAP@40 avg** |  | 0.434 |  | 0.419 |  | 0.404 |  | 0.400 |  | 0.40 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Exp 6** | | **Exp 7** | | **Exp 8** | | **Exp 9** | | **Exp 10** | |
| body\_w | **0.7** | | **0.7** | | **0.9** | | **0.9** | | **0.2** | |
| title\_w | **0.5** | | **0.7** | | **0.9** | | **0.9** | | **0.9** | |
| anchor\_w | **0.15** | | **0.15** | | **0.15** | | **0.8** | | **0.9** | |
|  | **Time**  **avg** | **MAP@40**  **avg** | **Time**  **avg** | **MAP@40**  **avg** | **Time**  **avg** | **MAP@40**  **avg** | **Time**  **avg** | **MAP@40**  **avg** | **Time avg** | **MAP@40**  **avg** |
| **Group1** | 3.65 | 0.344 | 3.68 | 0.351 | 3.64 | 0.351 | 3.64 | 0.513 | 3.725 | 0.516 |
| **Group2** | 5.25 | 0.413 | 5.25 | 0.411 | 5.19 | 0.406 | 5.21 | 0.434 | 5.28 | 0.462 |
| **Group3** | 3.16 | 0.523 | 3.20 | 0.519 | 3.13 | 0.523 | 3.14 | 0.523 | 3.33 | 0.540 |
| **Group4** | 1.92 | 0.277 | 1.94 | 0.285 | 1.91 | 0.280 | 1.90 | 0.269 | 1.94 | 0.265 |
| **Time avg** | 3.50 |  | 3.52 |  | 3.47 |  | 3.47 |  | 3.57 |  |
| **MAP@40 avg** |  | 0.389 |  | 0.392 |  | 0.39 |  | 0.434 |  | 0.446 |

From the results we have decided to continue with the parameters form experiment 10, as the average retrieval time didn’t change significantly between the experiments, this experiment provide us the best average MAP@40. After we add the group our model provide average retrieval time of 3.71 and average MAP@40 of 0.418.

For the query “LinkedIn” our engine gives the best results with MAP@40 = 0.998 and retrieval time of 0.146 sec. Our model retrieves the following results for this query:

[970755,"LinkedIn"], [ 50191962,"Timeline of LinkedIn"],[36070366, "2012 LinkedIn hack"],[41726116, "LinkedIn Learning"],[27769500,"LinkedIn Pulse"],[62976368,"HiQ Labs v. LinkedIn], [57147095, "LinkedIn Top Companies"],[63641225, "Ryan Roslansky"],[55679006, "Eric Ly"],[57636793,"Kevin Scott (computer scientist)"]

For the query “What is the best place to live in?” we get MAP@40 = 0.032 and retrieval time of 11.09. [18265302,"Best of Live (1996\u20132005)"],[18705980, "A Place to Live"],[50195220, "The Very Best of Santana \u2013 Live in 1968"],[33353799, "Best of Live (Bajaga i Instruktori album)"],[2661662,"Live All Over the Place"],[65961621, "The Last Best Place"],[1701281, "Best of Toadies: Live from Paradise"],[58767381, "The Best Place to Be"],[25070118, "MTV Europe Music Award for Best Live Act"],[39835477, "Setlist: The Very Best of New Riders of the Purple Sage Live"]

This probably occurs due to the query length, if the query length increases like in query “What is the best place to live in” our models precision starts to reduce. Also, we observed that this starts to happen with another long queries. To fix this and to improve our model precision probably we need to combine between more factors like to add the weight of the page ranks for each document and/or use the page views, and not only the combination between weights of the body, title and anchor. Also, other model like Word2Vec of using a neuronic grid probably can improve the precision of retrieved results.