# Information Retrieval H/M

Level4 - Dominik Bladek – 2144751b - Exercise 2

February 2019

Q1.

**Table 1: Performances of LTR vs PL2 on HP04 Topic Set**

| Measurement→ | MAP | P@5 |
|---|---|---|
| ↓Configuration | | |
| PL2 | 0.2251 | 0.0693 |
| LTR (PL2 Sample) | 0.469 | 0.1333 |

LTR vs PL2 on MAP: The *t*-value is 4.11011. The *p*-value is .000065. The result is significant at $p < .05$.

LTR vs PL2 on P@5: The *t*-value is 3.45342. The p-value is .000722. The result is significant at $p < .05$.

LTR is better than PL2 by a statistically significant margin on both used effectiveness metrics.

Q2a.

I have chosen min_dist (DSM1) and avg_dist (DSM2) to be my two proximity features. The main intuition behind min_dist is that if any occurrence of term a is close to any occurrence of term b, it indicated a relatedness between the terms. Avg_dist on the other hand sums every possible combination of distance between a and b and averages the result. This measure will promote terms that consistently occur close to one another in a localised area. I believe both features will improve the performance of the LTR by a statistically significant amount.

Q2b.

The initial implementation is very similar for both features. They firstly check and store postings that are ok to use. Afterwards they are casted to BlockPostings. Next, I have designed a function that calculates the minimum distances between the term positions and stores them. For the avg_dist feature, this part stores the average distance between the term positions. Lastly my first design choice is made as I aggregate the mean function score over all pairs of query terms.

The main difficulty in implementing the features was understanding how to use terrier “API” callouts correctly. Once this difficulty has been overcome, understanding the features themselves have become a difficulty. This problem has been solved by drawing the process on paper and then implementing it into code. Unit tests were a great way of making sure that the code was running correctly. The unit tests have been modified to check the results for various queries and all have successfully passed.

Q2c.

The unit tests focus on testing the proximity search features, especially the part that makes them different from each other. For the min_dist, as mentioned before, the unit tests focus on checking whether closest occurrence of a to b was found. For the avg_dist the unit tests focus on checking if

the feature sums up all the distances between a and b and if the average of those is returned. The avg_dist test did initially fail as a loop was not correctly set up and it was identified thanks to the test.

Q3.

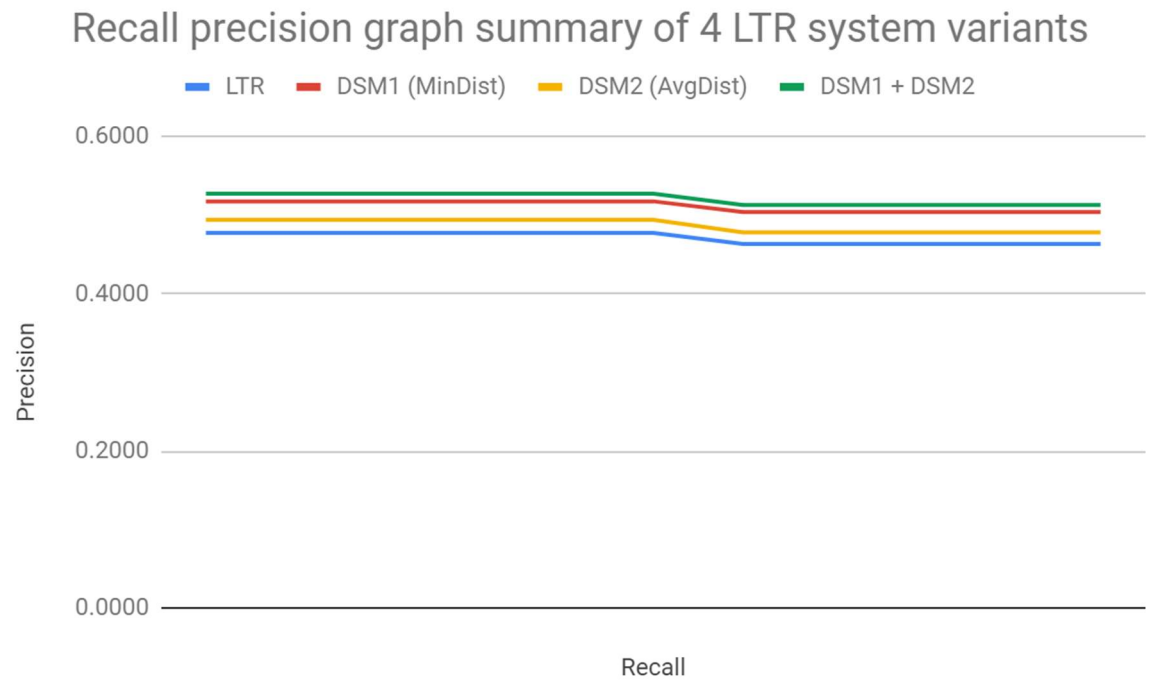
**Figure 2: Performances of LTR vs LTR with DSMs on HP04 Topic Set**

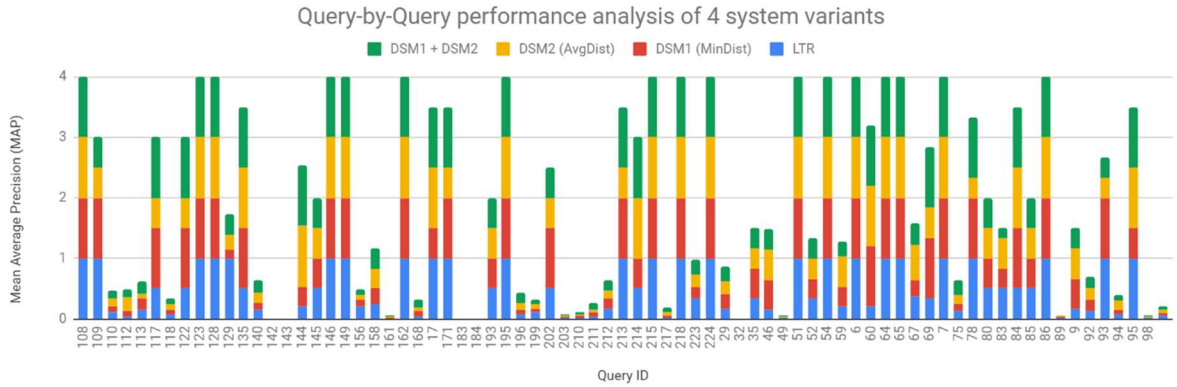
| Measurement→ | MAP | P@5 |
|---|---|---|
| ↓Configuration | | |
| **LTR (baseline)** | 0.4690 | 0.1333 |
| **LTR + DSM1** | 0.5092 | 0.1360 |
| **LTR + DSM2** | 0.4836 | 0.1413 |
| **LTR + DSM1 + DSM2** | 0.5187 | 0.1440 |

LTR vs LTR + DSM1 on MAP: The t-value is 0.5966. The p-value is .551689. The result is not significant at $p < .05$.
LTR vs LTR + DSM2 on MAP: The t-value is 0.2239. The p-value is .823143. The result is not significant at $p < .05$.
LTR vs LTR + DSM1 + DSM2 on MAP: The t-value is 0.73978. The p-value is .460607. The result is not significant at $p < .05$.

LTR + DSM1 is better by a statistically insignificant margin than LTR on MAP.
LTR + DSM2 is better by a statistically insignificant margin than LTR on MAP.
LTR + DSM1 + DSM2 is better by a statistically insignificant margin than LTR on MAP.

Q4.






| | Query-by-Query Comparison | | |
|---|---|---|---|
| | DSM1 vs LTR | DSM2 vs LTR | DSM1 + DSM2 vs LTR |
| Improved | 22 | 27 | 29 |
| Degraded | 17 | 13 | 9 |
| Unaffected | 36 | 35 | 37 |

| Improved Queries | | | | |
|---|---|---|---|---|
| Query ID | LTR | DSM1 (MinDist) | DSM2 (AvgDist) | DSM1 + DSM2 |
| 110 | 0.0418 | 0.0855 | 0.2292 | 0.1268 |
| 129 | 0.5 | 1 | 1 | 1 |
| 162 | 0.0357 | 0.1 | 0.0476 | 0.1429 |

| Harmed Queries | | | | |
|---|---|---|---|---|
| Query ID | LTR | DSM1 (MinDist) | DSM2 (AvgDist) | DSM1 + DSM2 |
| 128 | 1 | 0.1429 | 0.25 | 0.3333 |
| 196 | 0.1111 | 0.0556 | 0.0714 | 0.0833 |
| 86 | 0.0244 | 0.0156 | 0.0108 | 0.0137 |

Q5.

Learning-to-rank (LTR) is a recent paradigm used by commercial search engines to improve retrieval effectiveness by combining different features. We can see from the start of this paper that that the LTR using the PL2 weighing model is better than just the PL2 weighing model by a statistically significant margin. Furthermore, I explored whether adding features proposed in the research paper [1] further improved the LTR paradigm. The result of adding these features separately and together proved to be better than just the LTR paradigm using the PL2 weighing model but not by a statistically significant margin. We must remember that the initial LTR paradigm was compared to just the PL2 weighing model. This means that each one of the features added to the LTR paradigm proved to be even better than the base LTR. Additionally, in our case, we can see that combining both chosen features had a much greater positive effect than using the features separately.

Bibliography:

[1] http://ir.dcs.gla.ac.uk/~ronanc/papers/cumminsSIGIR09.pdf