Andrei-Mihai Nicolae
2147392n
17 February 2017

Information Retrieval

Assessed Exercise - Part I

Q1

Number of Indexed Documents	Size of Vocabulary	Number of Tokens	Number of Pointers
807775	2043788	572916194	177737957

Q2

The formula used in my Java code is the one included in the lecture slides as well:

```
double n = documentFrequency + 0.5;
double d = numberOfDocuments - documentFrequency + 0.5;
return tf * Math.log(d/n);
```

This is the corresponding mathematical formula:

$$w_{kd} = f_{kd}\left(\log\frac{(N - D_k) + 0.5}{D_k + 0.5}\right)$$

In the model shown, we are computing the weight of the kth term in the document (that is wkd). In order to achieve that, we are using its frequency in the document (fkd), the number of documents where the term appears (Dk) as well as the number of documents (N). This model also avoids the case where Dk could be 0, thus we add 0.5 to it.

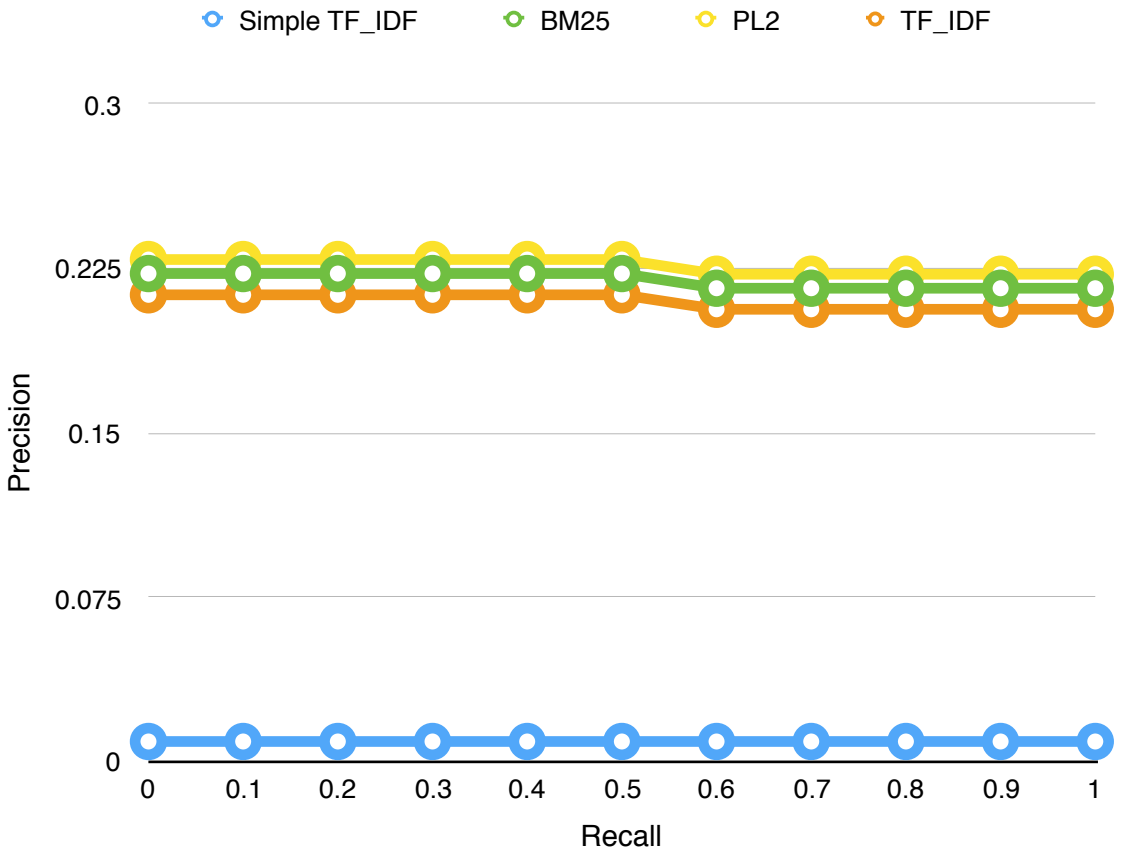
Q3

The following graph shows performance across various weighting models:

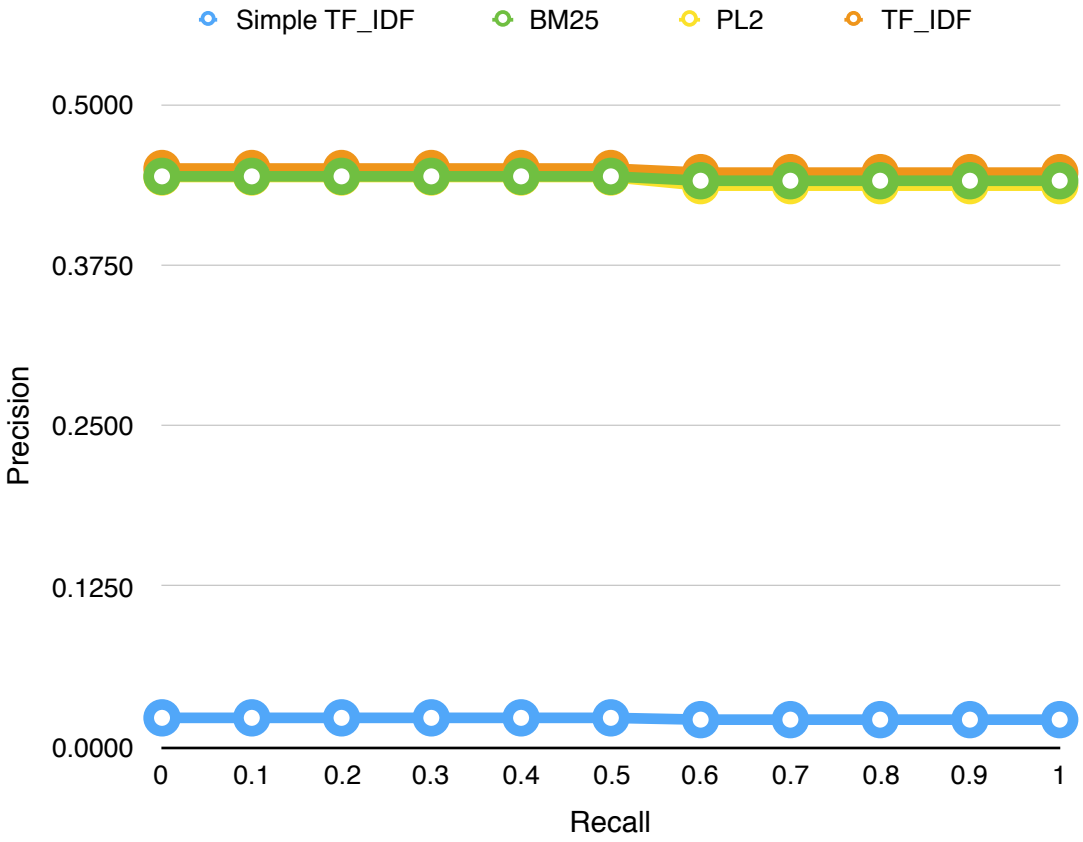
The Mean Average Precision over the 3 topics for different models

	HP_04	NP_04	TD_04	AM
Simple TF-IDF	0.0093	0.0223	0.0078	0.0131
TF-IDF	0.2089	0.4477	0.0698	0.2421
BM25	0.2186	0.4416	0.0703	0.2435
PL2	0.2251	0.4392	0.0695	0.2446

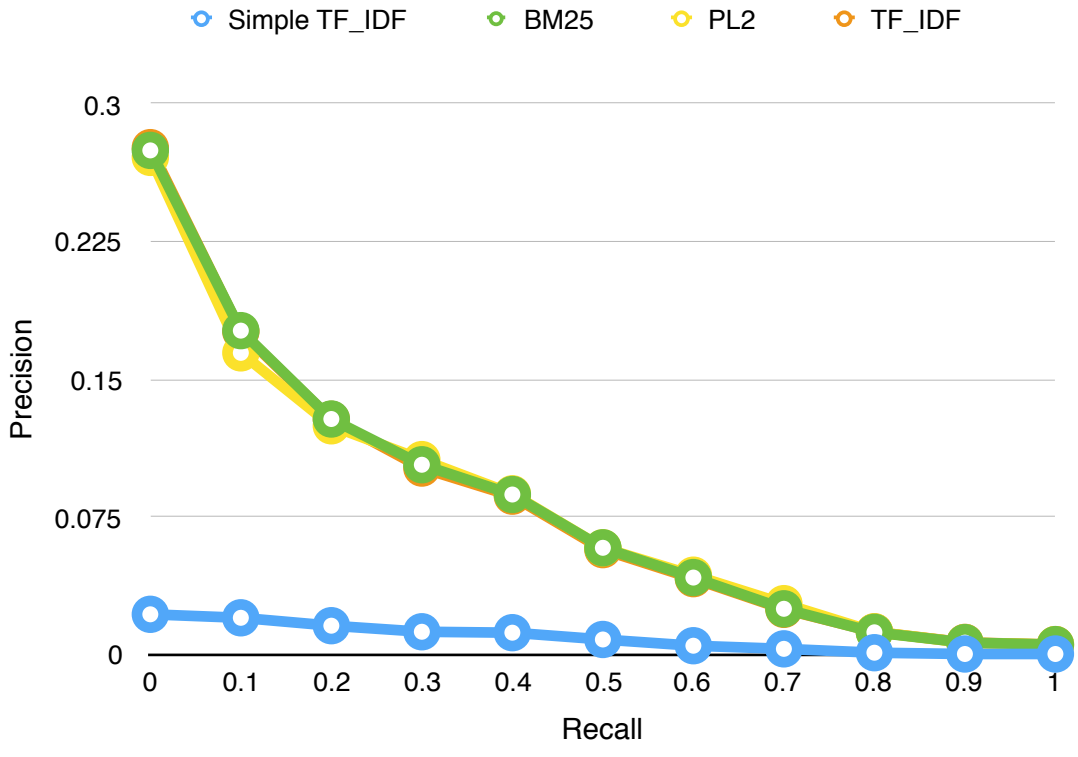
HP04 Precision-Recall Graph



NP04 Precision-Recall Graph



TD04 Precision-Recall Graph



As we can see in the graphs below, the simple TF_IDF that was implemented for Question 2 is far less performant than the other three competitors provided by Terrier.

The differences between BM25, TF_IDF and PL2 are very small, however the minor discrepancies that we can notice are:

- In **topic distillation**, it is extremely hard to distinguish which performed better; however, the winner seems to be **BM25**
- In **name page** findings, **TF_IDF** looks like the most performant model
- In **homepage** findings, **PL2** was above the other two

Even though the differences seem extremely small, after careful calculations, it appears that **PL2** is the most performant weighting model overall, having an MAP of 0.244, followed by BM25 and TF_IDF (0.243 and 0.242).

The conclusion that needs to be drawn here is, however, that a simple model will never be more performant than the more complex, well-built models such as the ones provided by the Terrier platform.

Q4