Information Retrieval

The BM25 Model

Dr. Seán Russell

School of Computer Science,
University College Dublin

Week 6

Table of Contents

Section Contents

- The "BM" in BM25 has the meaning "Best Match"
- A number of different BM rankings were developed through a series of experiments
- We will focus on the BM25 model, but will also reference some earlier variants
  - Such as BM15 & BM11
- Today, BM25 is considered to be a state-of-the-art retrieval method that operates using the same principles as TF-IDF but generally performs better than the classic version we have already studied

- The development of BM25 is based on the belief that good term weighting comes from three principles

1. Inverse document frequency (IDF)
2. Term frequency (TF)
3. Document length normalisation

Section Contents

- Inverse document frequency is a way of ranking how important a term is in a document collection
- If the term is contained in many/most of the documents, then searching using that term will be less useful
- If the term is contained in only a small number of documents, then searching using that term will be effective

- The following formula is used to calculate inverse document frequency in BM models for a term k_i

$$\log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

- Where:
  - N is the total number of documents in the collection
  - n_i is the total number of documents that the term k_i appears in
- These values are summed for all terms in both the document and the query

n_i \ N	50	75	100	200	400	800
1	5.04	5.63	6.05	7.06	8.06	9.06
2	4.28	4.88	5.30	6.31	7.32	8.32
4	3.37	3.99	4.42	5.45	6.46	7.47
8	2.32	2.99	3.44	4.50	5.53	6.54
10	1.95	2.64	3.11	4.18	5.22	6.23
25	0.00	0.99	1.57	2.78	3.88	4.93
50	-6.66	-0.99	0.00	1.58	2.80	3.89
75		-7.24	-1.57	0.73	2.11	3.26
100			-7.65	0.00	1.58	2.80
200				-8.65	0.00	1.58

n_i \ N	50	75	100	200	400	800
10	1.95	2.64	3.11	4.18	5.22	6.23
25	0.00	0.99	1.57	2.78	3.88	4.93
50	-6.66	-0.99	0.00	1.58	2.80	3.89
75		-7.24	-1.57	0.73	2.11	3.26
100			-7.65	0.00	1.58	2.80
200				-8.65	0.00	1.58

- Note: If a term is in 50% of the documents, then IDF is 0
- If a term is in more than 50% of the documents, then IDF is negative

- If a term is actually in our query, then we don't want a negative IDF
- We have 2 options:
  - Treat all terms in 50% of the documents or more as stopwords and do not index them
  - Modify the formula, e.g. the Lucene open source IR library adjusts it to

$$\log\left(1 + \frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

Section Contents

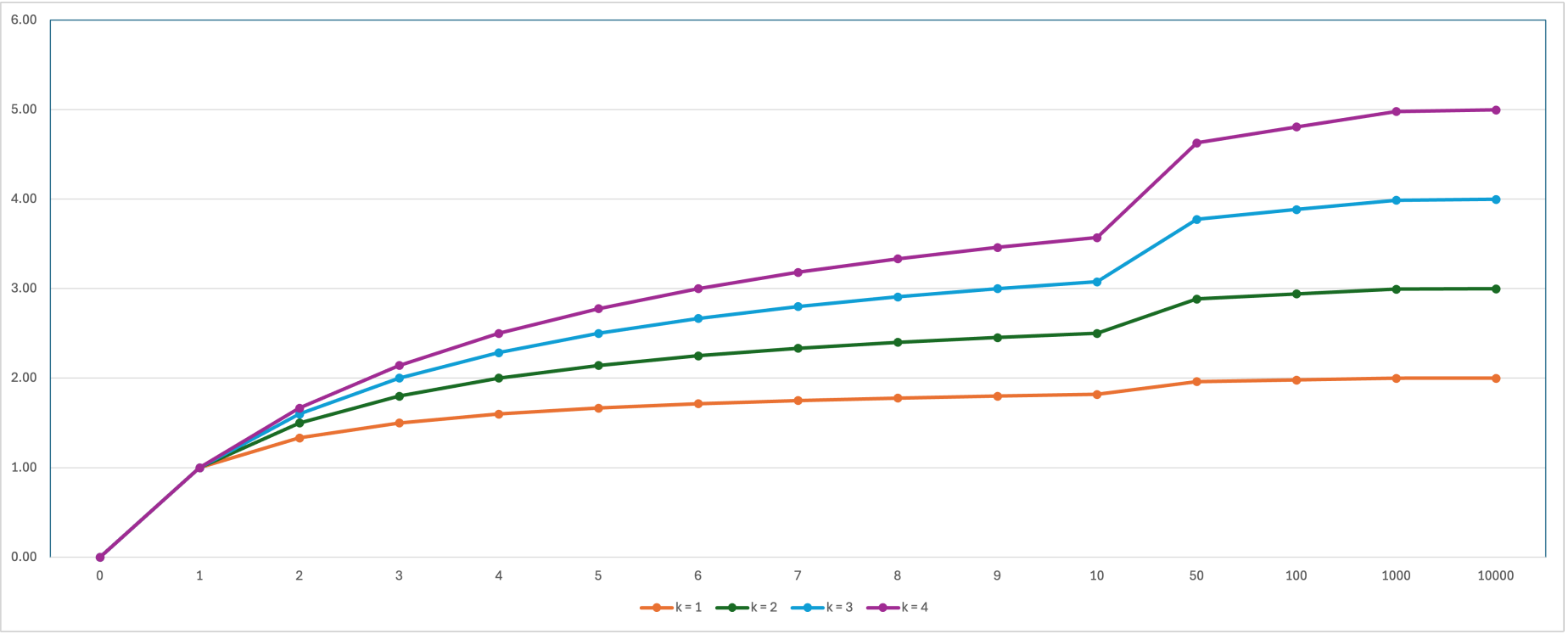
- Term frequency is usually shown as $(f_{i,j})$
  - Where i is the number of the term and j is the number of a document
- It is effectively a **count** of the number of times the term appears in the document
- BM models use a constant in a formula for normalising across different length documents

$$\frac{(k+1)f_{i,j}}{f_{i,j}+k}$$

- This formula prevents **saturation**
  - Saturation is where a term appears so frequently in a document that it receives a higher rank than it should

$$\frac{(k+1)f_{i,j}}{f_{i,j}+k}$$

- The constant k influences how quickly saturation is reached
- Result is between 0 and $k+1$



- The result of the calculation increases at different rates as the frequency of the term increases

- Another nice feature of this term frequency calculation is that it rewards complete matches over partial ones
- Imagine we have the query "cat dog"
  - Assume they both have the same IDF
  - Assume that k $= 1$
- If a document contains both "cat" and "dog" once, each will add $1 * IDF$ to the ranking score
- If a document contains "cat" twice but does not contain "dog" it will add $1.33 * IDF$ for cat and $0 * IDF$ for dog
- A document that fully matches the query will be preferred over one that only partially matches

Section Contents

- Comparing documents of different lengths can cause problems
  - If a very short document mentions "elephant", it is probably about elephants
  - If a very long document mentions "elephant" once, it is probably not about elephants
- BM25 normalises the length of documents in calculations through a further modification of the term frequency calculation

$$B_{i,j} = \frac{(k+1)\, f_{i,j}}{k\left((1-b) + \frac{b \times len(d_j)}{avg\_doclen}\right) + f_{i,j}}$$

$$B_{i,j} = \frac{(k+1)\, f_{i,j}}{k\left((1-b) + \frac{b \times len(d_j)}{avg\_doclen}\right) + f_{i,j}}$$

- k is the same constant as before and has the same effect on term frequency
- b is a new constant that allows us to choose how important document length is
  - The range of values for b is $0 \leq b \leq 1$

$f_{i,j}$ \ d	100	200	400	800	1600	3200	6400
1	1.43	1.29	1.08	0.82	0.55	0.33	0.18
2	1.67	1.57	1.40	1.16	0.86	0.57	0.34
3	1.76	1.69	1.56	1.35	1.06	0.75	0.47
4	1.82	1.76	1.65	1.47	1.20	0.88	0.58
5	1.85	1.80	1.71	1.55	1.31	1.00	0.67
6	1.88	1.83	1.75	1.61	1.39	1.09	0.76
7	1.89	1.85	1.78	1.66	1.45	1.16	0.83
8	1.90	1.87	1.81	1.69	1.50	1.23	0.90

- Here $k = 1$, $b = 0.75$ and $avg\_doclen = 500$

$$B_{i,j} = \frac{(k+1)\, f_{i,j}}{k\left((1-b) + \frac{b \times len(d_j)}{avg\_doclen}\right) + f_{i,j}}$$

- When $b = 0$, this becomes

$$B_{i,j} = \frac{(k+1) f_{i,j}}{f_{i,j} + k}$$ [1]

- When $b = 1$, this becomes

$$B_{i,j} = \frac{(k+1)\, f_{i,j}}{k\left(\frac{len(d_j)}{avg\_doclen}\right) + f_{i,j}}$$ [2]

[1] This is the formula used in BM15

[2] This is the formula used in BM11

- Not all document collections are the same
- So the ranking calculation should not be the same for every document collection
- BM25 provides two constants that we can tweak to improve performance
  - Changing the constant k changes how quickly increases in the term frequency stop causing increases in the results
  - Changing the constant b changes how important document length is in our calculation

- When we set the value of b to 0, then the calculation does not consider document length at all
- When we set the value of b to 1, then we use the full document length adjustment
- Typically, a value between these extremes is chosen (such as 0.75)

Section Contents

$$sim_{BM25}(d_j, q) \sim$$
$$\sum_{k_i \in d_j \wedge k_i \in q} \frac{(k+1) f_{i,j}}{k\left((1-b) + \frac{b \times len(d_j)}{avg\_doclen}\right) + f_{i,j}} \times \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

- This calculates the similarity score between a document d_j and a query q using BM25
  - $k_i \in d_j \wedge k_i \in q$ means that this is applied to all terms k_i where that term is in both the document and the query

- **Q:** "President Lincoln" $k_0 = \text{"}president\text{"}, k_1 = \text{"}lincoln\text{"}$
- 500000 documents $N = 500000$
- "president" in 40000 documents $n_0 = 40000$
- "lincoln" in 300 documents $n_1 = 300$
- "president" occurs 15 times in d_{123} $f_{0,123} = 15$
- "lincoln" occurs 25 times in d_{123} $f_{1,123} = 25$
- d_{123} is 90% of the length of the average $\frac{len(d_{123})}{avg\_doclen} = 0.9$
- $k = 1, b = 0.75$

- Calculation for k_0 ("president") in d_{123}

$$\frac{2 \times 15}{0.25 + 0.75 \times 0.9 + 15} \times log\left(\frac{500000 - 40000 + 0.5}{40000 + 0.5}\right)$$

$$\frac{30}{15.925} \times log\left(\frac{460000.5}{40000.5}\right)$$

$$= 1.88 \times 3.52 = 6.6378$$

- Calculation for k_1 ("lincoln") in d_{123}

$$\frac{2 \times 25}{0.25 + 0.75 \times 0.9 + 25} \times log\left(\frac{500000 - 300 + 0.5}{300 + 0.5}\right)$$

$$\frac{50}{25.925} \times log\left(\frac{499700.5}{300.5}\right)$$

$$= 1.93 \times 10.70 = 20.6355$$

- Result is sum of these values

$$6.6378 + 20.6355 = 27.2732$$

- **Q:** "President Lincoln" $k_0 = \text{"}president\text{"}, k_1 = \text{"}lincoln\text{"}$
- 500000 documents $N = 500000$
- "president" in 40000 documents $n_0 = 40000$
- "lincoln" in 300 documents $n_1 = 300$
- "president" occurs 43 times in d_7 $f_{0,7} = 43$
- "lincoln" occurs 4 times in d_7 $f_{1,7} = 4$
- d_7 is 85% of the length of the average $\frac{len(d_7)}{avg\_doclen} = 0.85$
- $k = 1, b = 0.75$

- Calculation for k_0 ("president") in d_7

$$\frac{2 \times 43}{0.25 + 0.75 \times 0.9 + 43} \times log\left(\frac{500000 - 40000 + 0.5}{40000 + 0.5}\right)$$

$$\frac{86}{43.8875} \times log\left(\frac{460000.5}{40000.5}\right)$$

$$= 1.96 \times 3.52 = 6.9046$$

- Calculation for k_1 ("lincoln") in d_7

$$\frac{2 \times 4}{0.25 + 0.75 \times 0.9 + 4} \times log\left(\frac{500000 - 300 + 0.5}{300 + 0.5}\right)$$

$$\frac{8}{4.8875} \times log\left(\frac{499700.5}{300.5}\right)$$

$$= 1.64 \times 10.70 = 17.5132$$

- Result is sum of these values

$$6.9046 + 17.5132 = 24.1478$$

- There are a number of variations of BM25
- Two of these are particularly notable
  - BM25F: Allows different fields to be given different importance in the document
    - E.g. Title, headlines, main text
  - BM25+: Addresses the issue where short documents would be given scores that are too high

- Most IR researchers agree that it outperforms the vector model on general collections
- It takes a lot of time and many, many experiments to reach this kind of consensus
  - Even though BM25 was originally proposed in the 1980s, the well-known Lucene IR library only adopted it as its default similarity score in 2016 when it replaced the Vector Space Model