Introduction to

Information Retrieval

Probabilistic Information Retrieval

tf-idf weighting

- The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$\mathrm{w}_{t,d} = \log(1 + \mathrm{tf}_{t,d}) \times \log_{10}(N / \mathrm{df}_t)$$

- Best known weighting scheme in information retrieval
  - Note: the "-" in tf-idf is a hyphen, not a minus sign!
  - Alternative names: tf.idf, tf x idf
- Increases with the number of occurrences within a document
- Increases with the rarity of the term in the collection

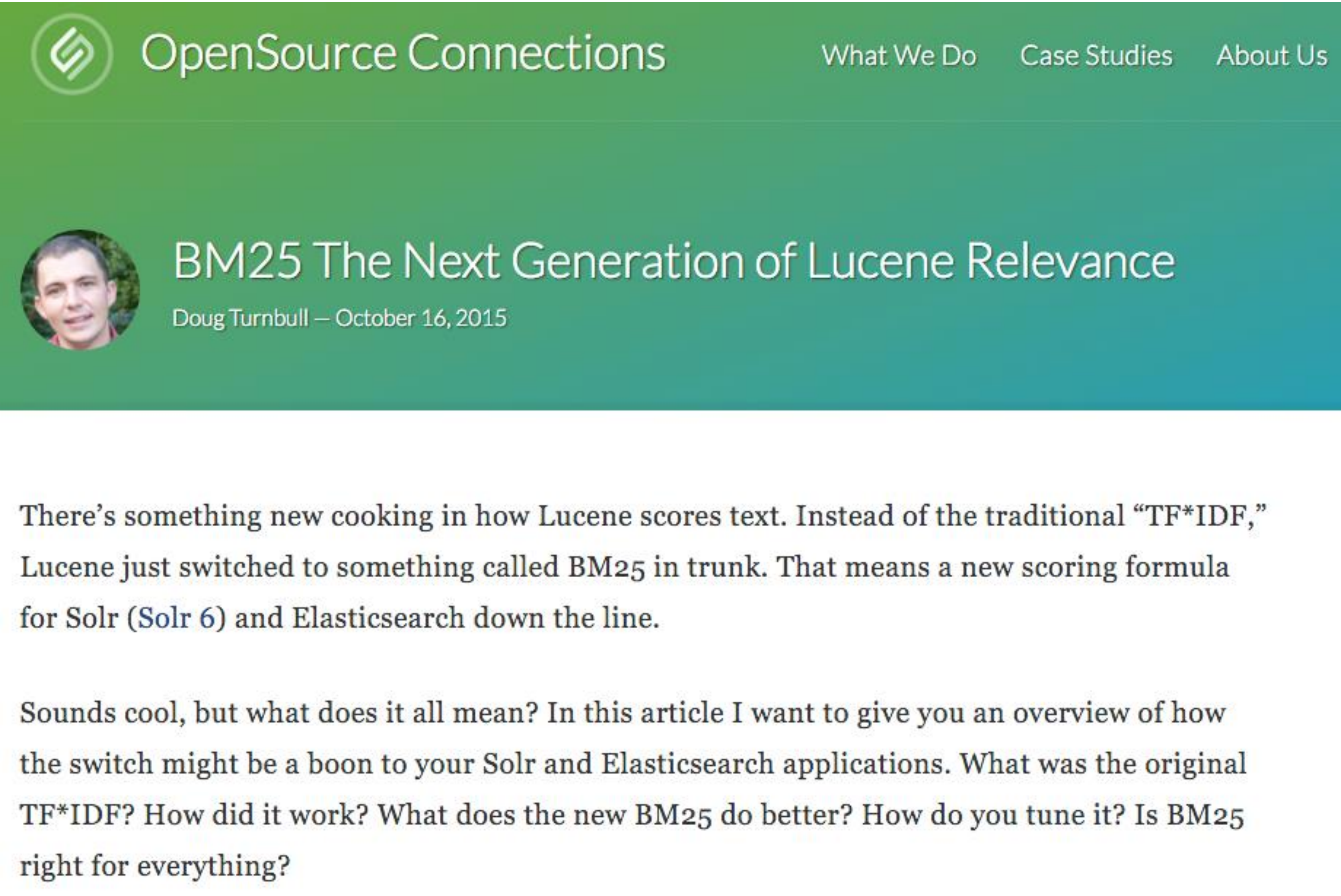
Binary → count → weight matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Each document is now represented by a real-valued vector of tf-idf weights $\in R^{|V|}$

$$\text{Score}(q,d) = \sum_{t \in q \cap d} \text{tf.idf}_{t,d}$$

6. BM25



BM25 The Next Generation of Lucene Relevance

Doug Turnbull — October 16, 2015

There's something new cooking in how Lucene scores text. Instead of the traditional "TF*IDF," Lucene just switched to something called BM25 in trunk. That means a new scoring formula for Solr (Solr 6) and Elasticsearch down the line.

Sounds cool, but what does it all mean? In this article I want to give you an overview of how the switch might be a boon to your Solr and Elasticsearch applications. What was the original TF*IDF? How did it work? What does the new BM25 do better? How do you tune it? Is BM25 right for everything?

Okapi BM25 [Robertson et al. 1994, TREC City U.]

- BM25 “Best Match 25” (they had a bunch of tries!)
  - Developed in the context of the Okapi system
  - Started to be increasingly adopted by other teams during the TREC competitions
  - It works well
- Goal: be sensitive to term frequency and document length while not adding too many parameters
  - (Robertson and Zaragoza 2009; Spärck Jones et al. 2000)

“Early” versions of BM25

- Version 1: using the saturation function

$$c_i^{BM25v1}(tf_i) = c_i^{BIM} \frac{tf_i}{k_1 + tf_i}$$

- Version 2: BIM simplification to IDF

$$c_i^{BM25v2}(tf_i) = \log \frac{N}{df_i} \text{´} \frac{(k_1+1)tf_i}{k_1 + tf_i}$$

  - (k_1+1) factor doesn’t change ranking, but makes term score 1 when $tf_i = 1$
  - Similar to *tf-idf*, but term scores are bounded

Document length normalization

- Longer documents are likely to have larger tf_i values

- Why might documents be longer?
  - Verbosity: suggests observed tf_i too high
  - Larger scope: suggests observed tf_i may be right

- A real document collection probably has both effects
- … so should apply some kind of partial normalization

Document length normalization

- Document length:

$$dl = \sum_{i \in V} tf_i$$

- *avdl*: Average document length over collection
- Length normalization component

$$B = \left((1 - b) + b \frac{dl}{avdl} \right) \qquad 0 \le b \le 1$$

  - $b = 1$ full document length normalization
  - $b = 0$ no document length normalization

Okapi BM25

- Normalize *tf* using document length

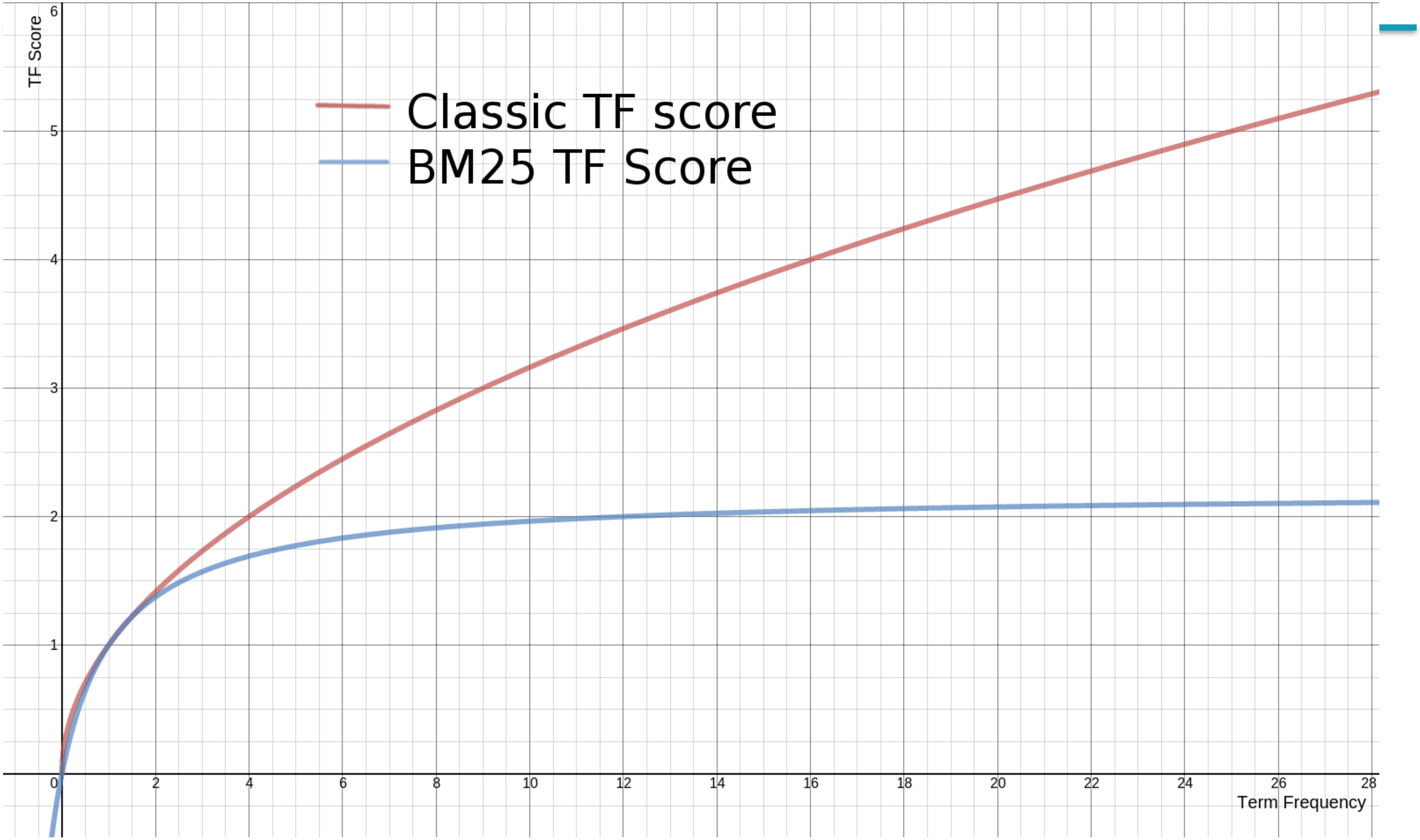
$$tf_i' = \frac{tf_i}{B}$$

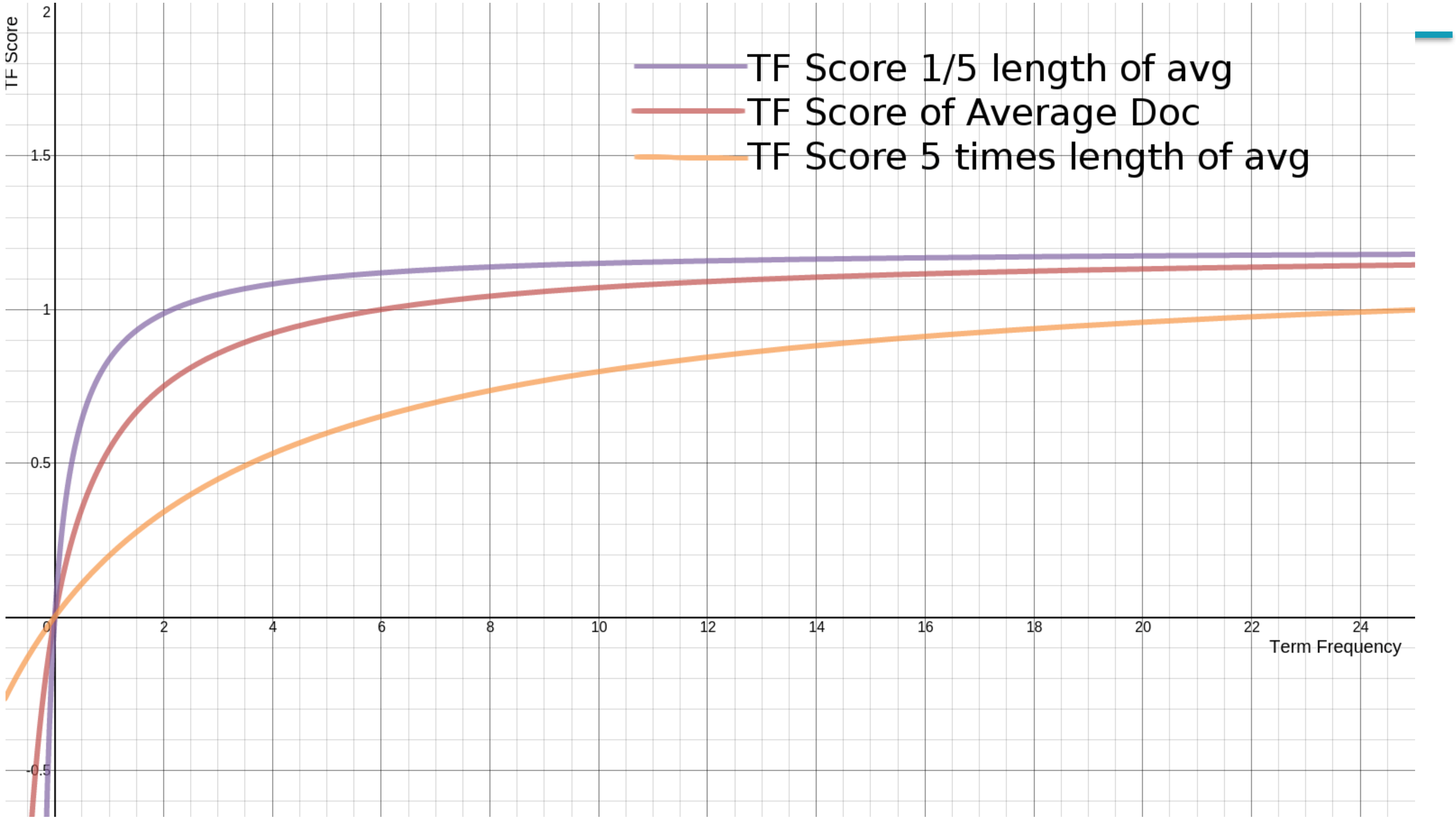
$$c_i^{BM25}(tf_i) = \log\frac{N}{df_i} \times \frac{(k_1+1)tf_i'}{k_1+tf_i'}$$

$$= \log\frac{N}{df_i} \times \frac{(k_1+1)tf_i}{k_1((1-b)+b\frac{dl}{avdl})+tf_i}$$

- BM25 ranking function

$$RSV^{BM25} = \sum_{i \in q} c_i^{BM25}(tf_i);$$


TF Score
6
5
4
3
2
1
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
Term Frequency
Classic TF score
BM25 TF Score


TF Score
2
1.5
1
0.5
-0.5
0
2
4
6
8
10
12
14
16
18
20
22
24
Term Frequency
TF Score 1/5 length of avg
TF Score of Average Doc
TF Score 5 times length of avg

Okapi BM25

$$RSV^{BM25} = \sum_{i \in q} \log \frac{N}{df_i} \times \frac{(k_1 + 1) tf_i}{k_1((1 - b) + b \frac{dl}{avdl}) + tf_i}$$

- k_1 controls term frequency scaling
  - $k_1 = 0$ is binary model; k_1 large is raw term frequency
- b controls document length normalization
  - $b = 0$ is no length normalization; $b = 1$ is relative frequency (fully scale by document length)
- Typically, k_1 is set around 1.2–2 and b around 0.75

Why is BM25 better than VSM tf-idf?

- Suppose your query is [machine learning]
- Suppose you have 2 documents with term counts:
  - doc1: learning 1024; machine 1
  - doc2: learning 16; machine 8
- tf-idf: $\log_2$ tf * $\log_2$ (N/df)
  - doc1: 11 * 7 + 1 * 10 = **87**
  - doc2: 5 * 7 + 4 * 10 = **75**
- BM25: $k_1 = 2$
  - doc1: 7 * 3 + 10 * 1 = **31**
  - doc2: 7 * 2.67 + 10 * 2.4 = **42.7**

Example

- Imagine we have two documents (A and B) and a query "protein-folding." We'll use the following parameters:
- **k1:** 1.2, **b:** 0.75, **avgdl:** 500
- **Document A:** Length (dl): 1000 words, Term frequency of "protein-folding" (tf): 10, idf of "protein-folding": 2
- **Document B:** Length (dl): 200 words, Term frequency of "protein-folding" (tf): 10, idf of "protein-folding": 2

Ranking with features

- Textual features
  - Zones: Title, author, abstract, body, anchors, …
  - Proximity
  - …
- Non-textual features
  - File type
  - File age
  - Page rank
  - …

Ranking with zones

- First combine evidence across zones for each term
- Then combine evidence across terms

BM25F with zones

- Calculate a weighted variant of total term frequency
- … and a weighted variant of document length

$$\tilde{tf}_i = \sum_{z=1}^{Z} v_z tf_{zi} \qquad \tilde{dl} = \sum_{z=1}^{Z} v_z len_z \qquad av\tilde{dl} = \text{Average } \tilde{dl} \text{ across all documents}$$

where

v_z is zone weight

tf_{zi} is term frequency in zone z

len_z is length of zone z

Z is the number of zones

Simple BM25F with zones

$$RSV^{SimpleBM25F} = \sum_{i \in q} \log \frac{N}{df_i} \times \frac{(k_1+1)t\tilde{f}_i}{k_1((1-b)+b\frac{d\tilde{l}}{avd\tilde{l}})+t\tilde{f}_i}$$

- **Example:** A document with "apple" in the title and "pie" in the body might be ranked higher for the query "apple pie" than a document with both terms only in the body.

- But we may want zone-specific parameters (*k_1*, *b*, IDF)

BM25F

- Empirically, zone-specific length normalization (i.e., zone-specific *b*) has been found to be useful

$$\tilde{tf}_i = \sum_{z=1}^{Z} v_z \frac{tf_{zi}}{B_z}$$

$$B_z = \left((1 - b_z) + b_z \frac{len_z}{avlen_z} \right), \quad 0 \le b_z \le 1$$

$$RSV^{BM25F} = \sum_{i \in q} \log \frac{N}{df_i} \cdot \frac{(k_1 + 1)\tilde{tf}_i}{k_1 + \tilde{tf}_i}$$

See Robertson and Zaragoza (2009: 364)

Resources

S. E. Robertson and K. Spärck Jones. 1976. Relevance Weighting of Search Terms. *Journal of the American Society for Information Sciences* 27(3): 129–146.

C. J. van Rijsbergen. 1979. *Information Retrieval.* 2nd ed. London: Butterworths, chapter 6. http://www.dcs.gla.ac.uk/Keith/Preface.html

K. Spärck Jones, S. Walker, and S. E. Robertson. 2000. A probabilistic model of information retrieval: Development and comparative experiments. Part 1. *Information Processing and Management* 779–808.

S. E. Robertson and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3(4): 333-389.