Introduction to

Information Retrieval

BM25 and BM25F

Slides borrowed from Stanford with slight modifications

OpenSource Connections

What We Do Case Studies About Us

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.]

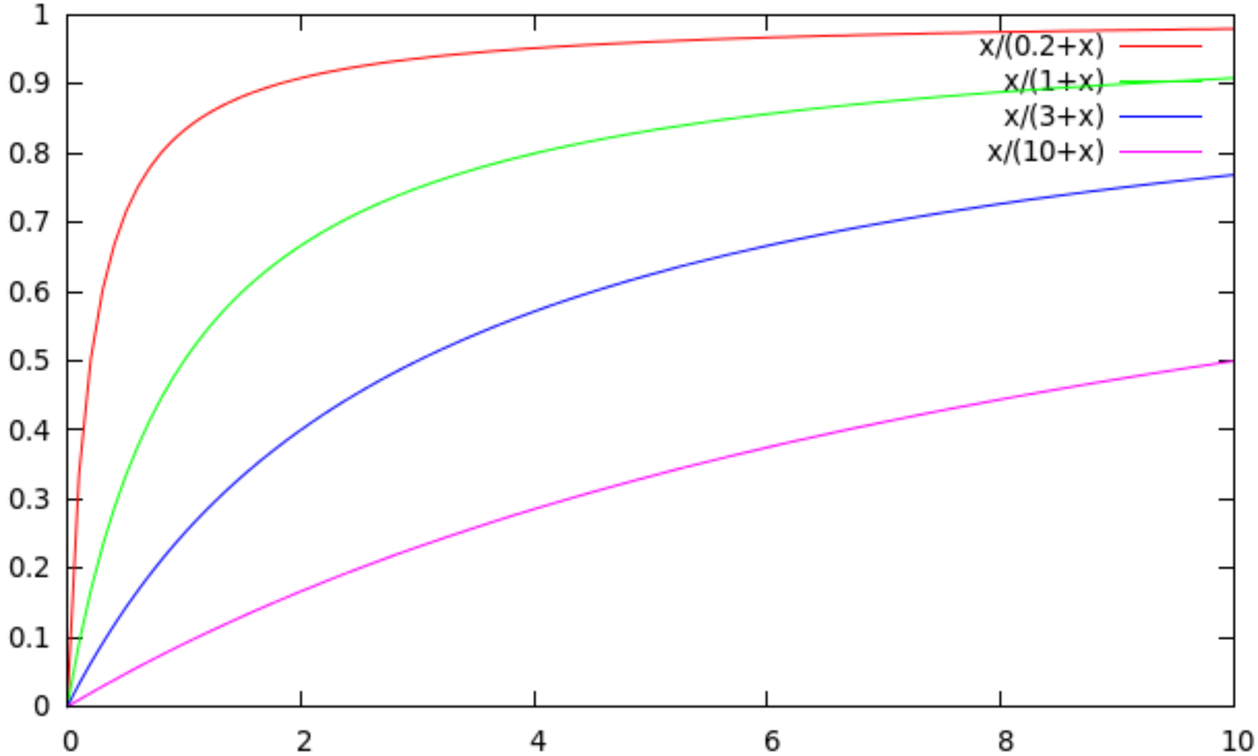
- BM25 “Best Match 25” (they had a bunch of tries!)
  - Developed in the context of the Okapi system
  - Started to be increasingly adopted by other teams during the TREC competitions
  - It works well
- Goal: be sensitive to term frequency and document length while not adding too many parameters
  -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

The saturation function

$$\frac{tf}{k_1 + tf}$$

- If *tf = 0, its value = 0*
- *Its value increases* monotonically with *tf*.
- … but asymptotically approaches a maximum value as $tf \to \infty$ [not true for simple scaling of *tf*]

Saturation function



- For high values of k_1, increments in tf_i continue to contribute significantly to the score
- Contributions tail off quickly for low values of k_1

“Early” version of BM25

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

- (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

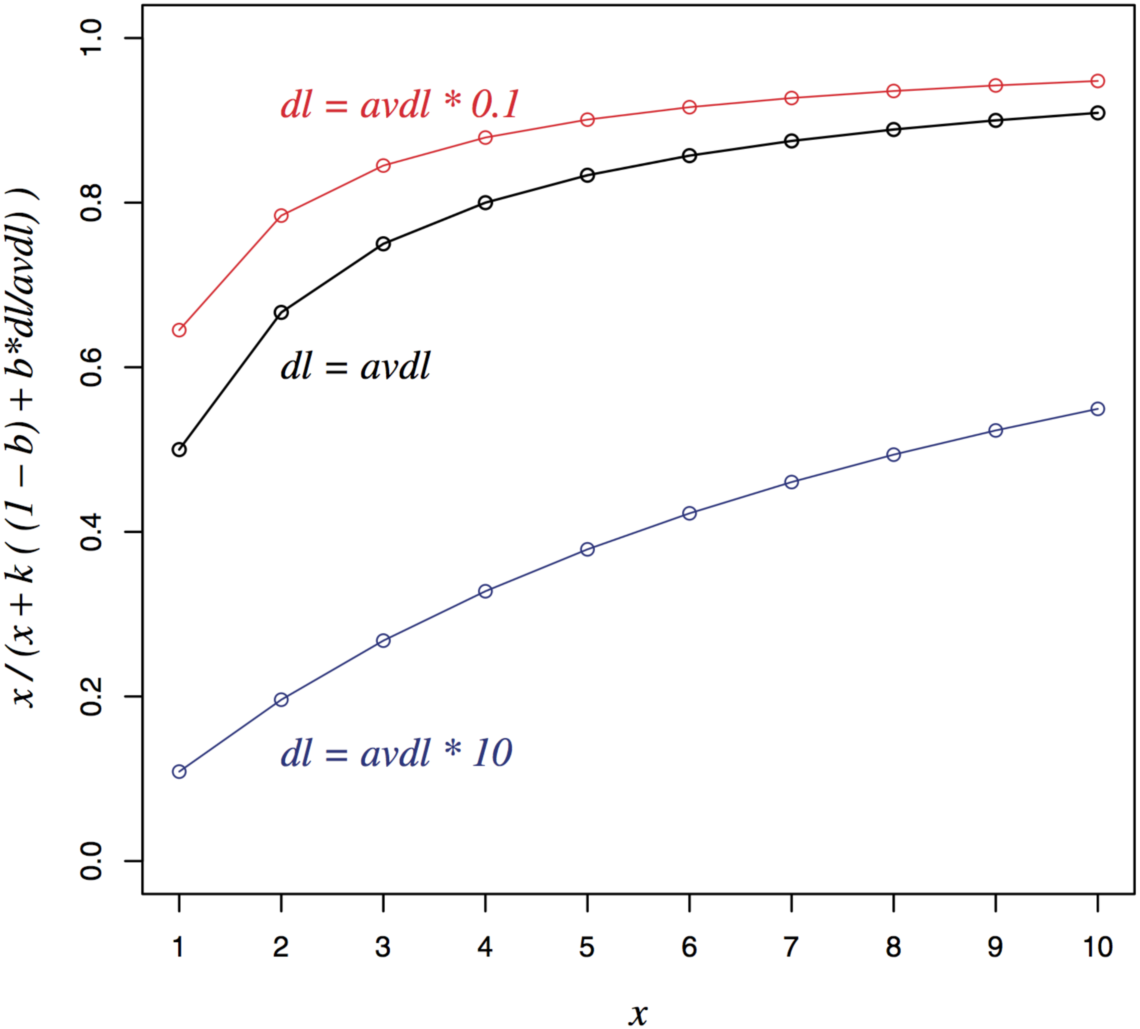
$$tf_i' = \frac{tf_i}{B}$$

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

- BM25 ranking function

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

Document length normalization



Okapi BM25

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

- 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
- *IIR* sec. 11.4.3 discusses incorporating query term weighting and (pseudo) relevance feedback

The BM25 formula

TF-IDF component for document TF component for query

$$rel(q,D)=\sum_{i=1}^{n}\boxed{IDF(q_i)\frac{tf_i(k_1+1)}{tf_i+k_1(1-b+b\frac{|D|}{avg|D|})}}\boxed{\frac{qtf_i(k_2+1)}{k_2+qtf_i}}$$

- b is usually set to [0.75]
- k_1 is usually set to [1.2, 2.0]
- k_2 is usually set to (0, 1000]

Vector space model with TF-IDF schema!

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: $(1+\log_2 \text{tf}) * \log_2 (N/df)$
  - doc1: 11 * 7 + 1 * 10 = **87**
  - doc2: 5 * 7 + 4 * 10 = **75**
- BM25: $k_1 = 2$, $(k_1+1)*\text{tf} / (k_1 + \text{tf}) * \log_2 (N/df)$
  - doc1: 7 * 3 + 10 * 1 = **31**
  - doc2: 7 * 2.67 + 10 * 2.4 = **42.7**

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

- Straightforward idea:
  - Apply your favorite ranking function (BM25) to each zone separately
  - Combine zone scores using a weighted linear combination
- But that seems to imply that the eliteness properties of different zones are different and independent of each other
  - …which seems unreasonable

Ranking with zones

- Alternate idea
  - Assume eliteness is a term/document property shared across zones
  - … but the relationship between eliteness and term frequencies are zone-dependent
    - e.g., denser use of elite topic words in title
- Consequence
  - 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

$$\tilde{tf}_i = \sum_{z=1}^{Z} v_z tf_{zi} \qquad \tilde{dl} = \sum_{z=1}^{Z} v_z len_z \qquad \tilde{avdl} =$$ Average $\tilde{dl}$ 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

$$RSV^{SimpleBM25F} = \sum_{i \in q} \log \frac{N}{df_i} \cdot \frac{(k_1 + 1)\tilde{tf}_i}{k_1((1-b) + b\frac{\tilde{dl}}{avd\tilde{l}}) + \tilde{tf}_i}$$

- Simple interpretation: zone z is "replicated" v_z times
- But we may want zone-specific parameters (b)

BM25F

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

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

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

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

See Robertson and Zaragoza (2009: 364)

Ranking with non-textual features

- Assumptions
  - Usual independence assumption
    - Independent of each other and of the textual features
  - Relevance information is ***query independent***
    - Usually true for features like page rank, age, type, …

Ranking with non-textual features

$$RSV = \sum_{i \in q} c_i(tf_i) + \sum_{j=1}^{F} \lambda_j V_j(f_j)$$

and λ_j is an artificially added free parameter to account for rescalings in the approximations

- Care must be taken in selecting V_j depending on *features*. E.g.

$$\log(\lambda'_j + f_j) \qquad \frac{f_j}{\lambda'_j + f_j} \qquad \frac{1}{\lambda'_j + \exp(-f_j \lambda''_j)}$$

- Explains why $RSV^{BM25} + \log(pagerank)$ works well

Resources

- S. E. Robertson and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3(4): 333-389.
- 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.
- T. Joachims. Optimizing Search Engines using Clickthrough Data. 2002. *SIGKDD*.
- E. Agichtein, E. Brill, S. Dumais. 2006. Improving Web Search Ranking By Incorporating User Behavior Information. 2006. *SIGIR*.