Practical Relevance Ranking for 11 Million Books, Part 3: Document Length Normalization.

Subscribe to Large-scale Search Blog

Submitted by Tom Burton-West on November 20, 2014

In Part 2 we argued that most relevance ranking algorithms used for ranking text documents are based on three fundamental features:

1. **Document Frequency:** the number of documents containing a query term.
2. **Term Frequency:** the number of times a query term occurs in the document.
3. **Document Length:** the number of words in the document.

This post discusses document length normalization.

Document length normalization is related to term frequency. Without length normalization, long documents would tend to get ranked above shorter documents even if the term in question is important in the short document but incidental to the topic of the long document.

In the example below, the 32 page book *Well Water Testing Guide* has 200 occurrences of the word "water." If we rank documents based on term frequency, we will use some function of the number of occurrences of query terms. For the query [water], without length normalization, long documents such as dictionaries and other reference books would be ranked higher than this book because they have more occurrences (higher term frequency) of the word "water".[i] In the chart below you can see that the *Engineering Index*, which contains around 3, 000 occurrences of the word "water", and the *American Encyclopaedic Dictionary,* which contains about 2,000 occurrences of the word "water", would be ranked much higher than the short document *Well Water Testing Guide*. This is because they have many more occurrences of the word "water" than the *Well Water Testing Guide*.

Title	Pages	Occurrences of "water"	Occurrences of "water" per page
"Well Water Testing Guide"	32	200	6.25

"Engineering Index"	2,000	3,000	1.5
"American Encyclopaedic Dictionary"	700	2,000	2.9

One simple length normalization formula is to divide the number of occurrences by the length of the document. For example, we can measure the length in pages and divide the number of occurrences (term frequency) by the number of pages as seen in Column 4 above. Dividing the number of occurrences by the number of pages increases the score of the 32 page document relative to the dictionary and index because it has many more occurrences per page of the word "water."

In practice, relevance ranking algorithms use more complex length normalization formulas, and there are more complex considerations regarding length normalization. In order to understand the details, it will be useful to discuss the history of length normalization in information retrieval.

History of Document Length Normalization

Early experiments with the SMART system in the 1970s and 1980s

In the early 1970's and 1980's Gerald Salton's group at Cornell did extensive experiments with the SMART information retrieval system. The SMART system is an implementation of the vector space model designed for experimentation. The SMART system was designed so that particular choices for handling term frequency, document frequency, and length normalization could be easily changed so that researchers could study and compare different combinations of these components. In a series of experiments based on twenty years of prior experimentation, Salton and Buckley (Salton and Buckley 1988) tested several different formulations of these three components. For example for the document frequency component (called collection frequency in the table below) they tried three different formulations:

$log\frac{N}{n}$,

$\log\frac{N-n}{n}$

and

1 (no idf/collection frequency component)

The table below shows the components and formulas used in their 1988 experiments[ii]. Note that inverse document frequency, idf, is called "collection frequency" in this early paper and in the chart below:

Table 1. Term-weighting components

Term Frequency Component		
b	1.0	binary weight equal to 1 for terms present in a vector (term frequency is ignored)
t	tf	raw term frequency (number of times a term occurs in a document or query text)
n	$0.5 + 0.5 \frac{\text{tf}}{\max \text{tf}}$	augmented normalized term frequency (tf factor normalized by maximum tf in the vector, and further normalized to lie between 0.5 and 1.0)
Collection Frequency Component		
x	1.0	no change in weight; use original term frequency component (*b*, *t*, or *n*)
f	$\log \frac{N}{n}$	multiply original tf factor by an inverse collection frequency factor (*N* is total number of documents in collection, and *n* is number of documents to which a term is assigned)
p	$\log \frac{N-n}{n}$	multiply tf factor by a probabilistic inverse collection frequency factor
Normalization Component		
x	1.0	no change; use factors derived from term frequency and collection frequency only (no normalization)
c	$1 \Big/ \sqrt{\sum_{\text{vector}} w_i^2}$	use cosine normalization where each term weight *w* is divided by a factor representing Euclidian vector length

Each particular formulation of a component is designated by a letter. This allows the experimenter to specify a particular combination of components using a series of letters. Since each of the three different components could be used for weighting documents or for weighting queries, each particular configuration of document and query weighting can be described by listing six letters such as "tfc*nfx." Salton and Buckley estimated that there were 1800 possible combinations of which 287 were unique. They tested these against six standard test collections in order to determine which combination was best.

The test collections and documents they used were both very small: 25-50 word abstracts with bibliographic data, bundled into collections of between 1,000 and 12,000 documents. The queries, however, were very long by modern web standards, averaging 10-20 words.[iii]

Salton and Buckley found that the best combination of components varied with the collection and types of queries. They attempted to describe how one could decide which combinations to use based on these characteristics. For example, some combinations worked better with very long queries. [iv]

At the time, operational information retrieval systems were several orders of magnitude larger than these test collections and various researchers suggested that a larger test collection was needed.[v]

The Text REtrieval Conference (TREC) in the 1990s

In 1992 the National Institute of Standards and Technology started the Text REtrieval Conference (TREC) with an initial collection of 2GB of text. At the time, this was a big challenge for existing systems.[vi] Document sizes in the TREC collections averaged around 300 words. However, a collection of relatively long (average 1,500 words) documents from the Federal Register was added to the TREC collections to serve as "noise" documents[vii] Collections were on the order of 500,000 documents. When the Text Retrieval Conference started up, systems had difficulty dealing with the huge (for the time) collections and the relatively large documents. As a result, over the first few years of the TREC conferences, various techniques were developed to deal with long documents.[viii]

Okapi at TREC and BM25

The Okapi information retrieval system of City University London did not do well with the "large" documents and "large" collections in TREC 1. As a result, between TREC 1 and TREC 3, the City University London team did a number of experiments with their relevance ranking algorithm (with at least 25 different versions of their ranking algorithm.) [ix] The eventual outcome of these experiments was that the core algorithm for the Okapi system (which was previously based only on idf) was modified to take term frequency and document length into account. [x] One of the findings of the experiments was that adding a tunable parameter to control the amount of length normalization improved retrieval results. In TREC 3 this parameter was added, and the version of the algorithm , which is known as BM25 (Best Match Algorithm Number 25), was first used. This algorithm has been widely used as a baseline algorithm in information retrieval experiments for the last two decades.[xi]

In adapting the Okapi algorithms to deal with long documents, Stephen Robertson articulated the "scope vs. verbosity" hypothesis:

We may postulate at least two reasons why documents might vary in length. Some documents may simply cover more material than others; an extreme version would have a long document consisting of a number of unrelated short documents concatenated together (the 'scope hypothesis"). An opposite view would have long documents like short documents but longer; in other words, a long document covers a similar scope to a short document, but simply uses more words (the "verbosity hypothesis").

Robertson et al. (1994a ,1994b p 235)

Robertson argued documents that are long because they are verbose should be normalized and documents that are long because they cover a wide scope should not be normalized. However, most documents are long because of a mix of scope and verbosity: *"The verbosity hypothesis suggests that we should simply normalise any observed tfs by dividing by document length. The scope hypothesis, on the other hand, at least in its extreme version, suggests the opposite. In a real collection of documents we will observe variations in length, which might be due to either effect, or a combination."* Robertson and Zaragoza (2009: p 359)

Since most documents are long because of a mix of scope and verbosity, Robinson's practical solution to this issue was to create a mixture model with a parameter "b" to control the mix of normalization vs no normalization.[xii] Below is the part of the BM25 formula that deals with length normalization:

$$w * \frac{1}{\left((1-b)+ b* \frac{dl}{avdl}\right)}$$

- w = document score before length normalization
- dl = document length.
- avdl = average document length.
- "b" is the tuning parameter that controls the mix of normalization/non-normalization to compensate for scope vs verbosity.
- "b" is a number between 0 and 1. Think of it as percentage normalization

If b is set to 1, then the equation becomes:

$$W * \frac{1}{\left(\frac{dl}{avdl}\right)} = W * \frac{avdl}{\mathrm{dl}}$$

This is 100% normalization and favors the verbosity hypothesis. If average document length were 100 words and you had a document with 200 words then the score would be multiplied by ½.

If b is set to 0, the equation just multiplies the weight by 1 so there is no normalization.

$$W * \frac{1}{\left((1-0)+ 0\frac{dl}{avdl}\right)} = W * \frac{1}{1} = W * 1$$

As implemented, the BM25 algorithm does not attempt to determine whether a particular document being ranked is long because it covers a wide scope or because it is verbose or any specific mix between the two. The same value of "b" is used for every document in a collection.

The "b" parameter of BM25 is tuned to general collection characteristics by testing it against a training collection of documents, queries, and relevance judgments that are similar to the collection to be used in production. Parameters are tweaked to produce the best results on some measure of relevance. In tests on various TREC collections and other IR research collections this strategy of tuning the algorithm has proven to be effective.

SMART catches up

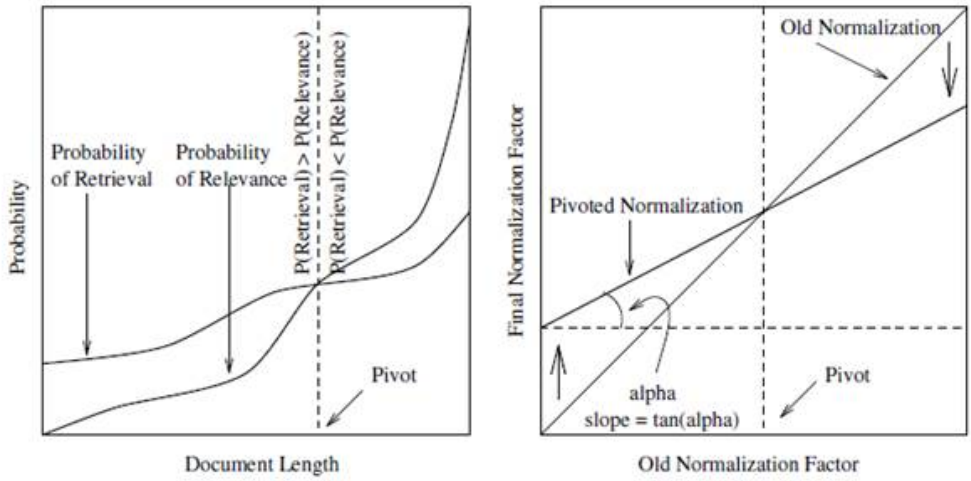
From the 1960s until TREC started in 1992, the SMART information retrieval system from Cornell was one of the leading systems and was generally used as the baseline system for testing new systems and algorithms. By the third TREC, the SMART system was not doing nearly as well as the Okapi system from City University.[xiii] Amit Singhal, a graduate student in Salton's group at Cornell (now senior VP of Google and the head of the Google core relevance ranking team) investigated why SMART was doing so

much worse than Okapi. Singhal determined that the comparatively poor performance of the SMART algorithms was due to Okapi's superior document length normalization. [xiv]

In his analysis, Singhal sorted documents that were judged relevant in the TREC collections by size and then put them into equal size bins. Using the median document size for each bin, he then graphed the probability of relevance. He discovered that the SMART system was doing too much length normalization (i.e. reducing the score based on the length of the document.) Thus as a result the SMART system was biased against long documents. The system was retrieving shorter documents with a probability greater than their probability of relevance and longer documents with a probability smaller than their probability of relevance.

In the figures below, the graph on the left shows a smoothed representation of the probability of relevance and probability of retrieval using SMART's default (cosine) term frequency normalization.[xv] In order to move the probability of retrieval closer to the probability of relevance, scores for documents smaller than a "pivot point" need to be reduced, and scores for documents longer than the pivot point need to be increased.

Singhal explained that the probability of retrieval is inversely proportional to the normalization factor. [xvi] The figure on the right shows how the normalization factor needs to be modified and demonstrates the "pivot. For documents shorter than the pivot point the normalization factor needs to be increased (which will decrease the scores of short documents), and for documents longer than the pivot point the normalization factor needs to be decreased (which will increase the scores of long documents.) Note that the slope of the line is controlled by a tunable parameter (alpha in the chart on the right).



Singhal did his doctoral thesis on length normalization (Singhal 1997). By using his "pivoted document weight normalization", the Cornell team got results comparable with the Okapi team.[xvii] Singhal's methodology has been replicated in a number of subsequent studies, and most groups using the vector space model have adopted Singhal's pivoted normalization method.

We suspect that the default Solr/Lucene ranking algorithm, which is loosely based on the vector space model, suffers from the same problem of ranking short documents too high and long documents too low. Robert Muir contributed a patch that implements "pivoted document length normalization" for Lucene. (https://issues.apache.org/jira/browse/LUCENE-2187) However, the patch has not been incorporated into Lucene, probably because the implementation of newer algorithms such as BM25, Language Models, Informaton-Based Models and Divergence from Randomness were considered a higher development priority.

Tuning Parameters

Both BM25 and "Pivoted Document Length Normalization" have parameters that need to be tuned to the characteristics of a document collection. More recent algorithms--such as the Divergence from Randomness (DFR) and Information-Based algorithms--also have tunable length normalization parameters.[xviii]

The values for these parameters are determined empirically by using a test collection of documents, queries, and relevance judgments, and doing a parameter sweep to optimize a particular retrieval effectiveness metric, such as mean average precision (MAP).[xix] There are several important points here:

1. The optimal parameter settings differ for different collections and may be impacted by characteristics of documents, queries, and relevance judgments[xx]
2. The values for the parameters are not determined by a theoretically justified method, nor are they determined based on some particular quantifiable property of the documents, queries or relevance judgments.[xxi]
3. Because the parameter settings are collection-specific, for optimal results they have to be empirically determined for each collection. In practice a test collection that is similar to the collection used in production is generally used.[xxii]

The empirical approach to tuning length normalization parameters using test collections has proven to be effective in tests on various TREC collections and other IR research collections. However, there remain many unanswered questions for a practitioner working with production systems:

1. Given a large production collection, such as the 11 million volume HathiTrust corpus, what criteria should be used for creating a representative test collection? [xxiii]
   A. How large does a test collection need to be compared to the target (production) collection? The INEX Book Track test collection contains about 50,000 books. Would results of experiments on the INEX Book Track test collection apply to a collection such as HathiTrust that is two orders of magnitude larger?
   B. What are the important properties of documents, queries, and relevance judgments that need to be similar between the production collection and the test collection?
2. Given limited resources, how can the relevance judgments needed for a test collection be collected at a reasonable cost?[xxiv]
3. How should length normalization be applied in different types of collections or different user tasks? Are there particular properties of documents, queries, or relevance judgments that might call for a different type of length normalization?

The TREC ad hoc collections used in almost all of the studies of length normalization (including both Robertson's Okapi work and Singhal's SMART work) make some specific assumptions about user needs and relevance that may not apply to different tasks or collections. The next post will discuss the assumptions used in the TREC ad hoc collections and their implications for relevance ranking of book-length documents.

References

Omar Alonso. 2013. Implementing crowdsourcing-based relevance experimentation: an industrial perspective. *Inf. Retr.* 16, 2 (April 2013), 101-120. DOI=10.1007/s10791-012-9204-1 http://dx.doi.org/10.1007/s10791-012-9204-1

Giambattista Amati. 2006. Frequentist and bayesian approach to information retrieval. In *Proceedings of the 28th European conference on Advances in Information Retrieval* (ECIR'06), Mounia Lalmas, Andy MacFarlane, Stefan Rüger, Anastasios Tombros, and Theodora Tsikrika (Eds.). Springer-Verlag, Berlin, Heidelberg, 13-24. DOI=10.1007/11735106_3 http://dx.doi.org/10.1007/11735106_3

Ben Carterette, Evangelos Kanoulas, and Emine Yilmaz. 2010. Low cost evaluation in information retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '10). ACM, New York, NY, USA, 903-903. DOI=10.1145/1835449.1835675 http://doi.acm.org/10.1145/1835449.1835675

Abdur Chowdhury, M. Catherine McCabe, David Grossman, and Ophir Frieder. 2002. Document normalization revisited. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '02). ACM, New York, NY, USA, 381-382. DOI=10.1145/564376.564454 http://doi.acm.org/10.1145/564376.564454

Norbert Fuhr. 2012. Salton award lecture: information retrieval as engineering science. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (SIGIR

'12). ACM, New York, NY, USA, 1-2. DOI=10.1145/2348283.2348285 http://doi.acm.org/10.1145/2348283.2348285 http://www.is.informatik.uni-duisburg.de/bib/pdf/ir/Fuhr_12.pdf

Norbert Fuhr (2010). IR Between Science and Engineering, and the Role of Experimentation. Keynote talk at CLEF 2010, Padua, Italy http://www.is.informatik.uni-duisburg.de/bib/pdf/talks/Fuhr_10t.pdf

Donna Harman. 2011. Information Retrieval Evaluation (1st ed.). Morgan & Claypool Publishers. http://www.morganclaypool.com/doi/abs/10.2200/S00368ED1V01Y201105ICR019

Ben He and Iadh Ounis. 2007. On setting the hyper-parameters of term frequency normalization for information retrieval. *ACM Trans. Inf. Syst.* 25, 3, Article 13 (July 2007). DOI=10.1145/1247715.1247719 http://doi.acm.org/10.1145/1247715.1247719

Timothy Jones, Andrew Turpin, Stefano Mizzaro, Falk Scholer, and Mark Sanderson. 2014. Size and Source Matter: Understanding Inconsistencies in Test Collection-Based Evaluation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (CIKM '14). ACM, New York, NY, USA, 1843-1846. DOI=10.1145/2661829.2661945 http://doi.acm.org/10.1145/2661829.2661945

Marijn Koolen and Jaap Kamps. The importance of anchor text for ad hoc search revisited. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York NY, USA, 2010. http://staff.science.uva.nl/~mhakoole/publications/2010/kool:impo10.pdf

Gabriella Kazai, Natasa Milic-Frayling, and Jamie Costello. 2009. Towards methods for the collective gathering and quality control of relevance assessments. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '09). ACM, New York, NY, USA, 452-459. DOI=10.1145/1571941.1572019 http://doi.acm.org/10.1145/1571941.1572019

David E. Losada, Leif Azzopardi, and Mark Baillie. 2008. Revisiting the relationship between document length and relevance. In *Proceedings of the 17th ACM conference on Information and knowledge management* (CIKM '08). ACM, New York, NY, USA, 419-428. DOI=10.1145/1458082.1458139 http://doi.acm.org/10.1145/1458082.1458139

David E. Losada and Leif Azzopardi. 2008. An analysis on document length retrieval trends in language modeling smoothing. *Inf. Retr.* 11, 2 (April 2008), 109-138. DOI=10.1007/s10791-007-9040-x http://dx.doi.org/10.1007/s10791-007-9040-x

Yuanhua Lv and ChengXiang Zhai. 2011. Adaptive term frequency normalization for BM25. In*Proceedings of the 20th ACM international conference on Information and knowledge management* (CIKM '11), Bettina Berendt, Arjen de Vries, Wenfei Fan, Craig Macdonald, Iadh Ounis, and Ian Ruthven (Eds.). ACM, New York, NY, USA, 1985-1988. DOI=10.1145/2063576.2063871 http://doi.acm.org/10.1145/2063576.2063871

Stephen Robertson. 2008. On the history of evaluation in IR. *J. Inf. Sci.* 34, 4 (August 2008), 439-456. DOI=10.1177/0165551507086989 http://dx.doi.org/10.1177/0165551507086989

ROBERTSON, S., WALKER, S., JONES, S., HANCOCK-BEAULIEU, M., AND GATFORD, M. 1993. Okapi at TREC-2. In Proceedings of the TREC-2 Conference (Gaithersburg, MD}, 21--25. http://research.microsoft.com/pubs/67648/okapi_trec2.pdf

ROBERTSON, S. E., WALKER, S., JONES, S., HANCOCK-BEAULIEU, M., AND GATFORD, M. 1994a. Okapi at TREC-3. In Proceedings of the TREC-3 Conference (Gaithersburg, MD), 109--128. http://research.microsoft.com/pubs/67649/okapi_trec3.pdf

S. E. Robertson and S. Walker. 1994b. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '94), W. Bruce Croft and C. J. van Rijsbergen (Eds.). Springer-Verlag New York, Inc., New York, NY, USA, 232-241.

Stephen Robertson and Hugo Zaragoza. 2009. *The Probabilistic Relevance Framework:BM25 and Beyond*. Now Publishers Inc., Hanover, MA, USA. http://www.soi.city.ac.uk/~ser/papers/foundations_bm25_review.pdf

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.* 24, 5 (August 1988), 513-523. DOI=10.1016/0306-4573(88)90021-0 http://dx.doi.org/10.1016/0306-4573(88)90021-0

M. Sanderson. Test collection based evaluation of information retrieval systems. Foundations and Trends in Information Retrieval, 4:247--375, 2010. http://www.seg.rmit.edu.au/mark/publications/my_papers/FnTIR.pdf

Amit Singhal, Chris Buckley, and Mandar Mitra. 1996a. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '96). ACM, New York, NY, USA, 21-29. DOI=10.1145/243199.243206 http://doi.acm.org/10.1145/243199.243206

Amit Singhal, Gerard Salton, Mandar Mitra, and Chris Buckley. 1996b. Document length normalization. *Inf. Process. Manage.* 32, 5 (September 1996), 619-633. DOI=10.1016/0306-4573(96)00008-8 http://dx.doi.org/10.1016/0306-4573(96)00008-8

Amit Singhal, Gerard Salton, Chris Buckley. 1996c. Length Normalization in Degraded Text Collections. Fifth Annual Symposium on Document Analysis and Information Retrieval, 149-162, 1996. http://singhal.info/ocr-norm.pdf

Amitabh Kumar Singhal. 1997. *Term Weighting Revisited*. Ph.D. Dissertation. Cornell University, Ithaca, NY, USA. UMI Order No. GAX97-14899.http://dspace.library.cornell.edu/handle/1813/7281 http://dspace.library.cornell.edu/bitstream/1813/7281/1/97-1626.pdf See also http://singhal.info/

Anne Schuth, Floor Sietsma, Shimon Whiteson, and Maarten de Rijke. Optimizing Base Rankers Using Clicks: A Case Study using BM25. In ECIR 2014: Proceedings of the Thirty-Sixth European Conference on Information Retrieval, pp. 75–87, April 2014. http://www.anneschuth.nl/wp-content/uploads/2014/01/ecir2014-schuth-bm25.pdf

K. Sparck Jones and C.J. van Rijsbergen, 1975. *Report on the need for and provision of an 'ideal' information retrieval test collection*, Computer Laboratory, University of Cambridge, 1975 (BP R&D Report 5266) http://sigir.org/files/museum/pub-14/pub_14.pdf

K. Sparck Jones, S. Walker and S.E. Robertson. 1998. *A probabilistic model of information retrieval : development and status*, Technical Report 446, Computer Laboratory, University of Cambridge, 1998. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-446.html, http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=27CD0692F09123B03EAB5D1F5BA23168?doi=10.1.1.12.5386&rep=rep1&type=pdf

K. Sparck Jones, S. Walker, and S. E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments Part 2. *Inf. Process. Manage.* 36, 6 (November 2000), 809-840. DOI=10.1016/S0306-4573(00)00016-9 http://dx.doi.org/10.1016/S0306-4573(00)00016-9

Ellen M. Voorhees and Donna K. Harman. 2005. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press.

Chengxiang Zhai and John Lafferty. 2001. A study of smoothing methods for language models applied to Ad Hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '01). ACM, New York, NY, USA, 334-342. DOI=10.1145/383952.384019 http://doi.acm.org/10.1145/383952.384019

Justin Zobel and Alistair Moffat. 1998. Exploring the similarity space. *SIGIR Forum* 32, 1 (April 1998), 18-34. DOI=10.1145/281250.281256 http://doi.acm.org/10.1145/281250.281256

---

[i] Idf or inverse document frequency does not come into play with one word queries.

[ii] Note that the weighting under tf labeled “n” could be considered a form of length normalization since it normalizes by the maximum term frequency.

[iii] With 10-20 word queries and 25-50 word documents it is easy to see why the vector space model essentially considers a query a document and compares the similarity between two documents (i.e. the

query document and the "real" document). With 10-20 word queries, taking into account the length and term frequency in queries also makes sense.

[iv] Ten years later Zobel and Moffat (1998) replicated Salton and Buckley's experiments with larger collections and quite a few more variables. They estimated there were about 100,000 unique combinations of parameters. They tested with three different types of queries and two different document collections. No measure worked best for all collections. They found it was difficult to predict which types of parameter combinations would work best for a particular combination of query type and collection. Another important finding was that "*...large differences in results can easily be the consequence of minor variations to the similarity measures such as the base of the logarithms, and whether, as in another case we encountered, the '+1' addition takes place before or after the logarithms are taken.*"

[v] As early as 1975 Karen Sparck-Jones and CJ Van Rijsbergen issued a report on the need for an "ideal" test collection (Sparck-Jones and vanRijsbergen 1975).http://http//sigir.org/files/museum/pub-14/pub_14.pdf

[vi] Harman (2011:p 28) notes that when TREC started up in 1992, the storage cost of 4GB of text (storage for the documents and the index to the documents) cost around $10,000.

[vii] Voohees and Harman (2005: pp. 25-26 and pp. 36-38) describe how the Federal Register documents were added as "noise" documents and Harman's analysis shows that the Federal Register contributed very few relevant documents compared to the other TREC sub-collections such as the newspaper and newswire sources.

[viii] See Voorhees and Harman 2005 pp. 84-86.

[ix] Robertson 1993 states that more than 20 combinations of weighting functions were implemented.

[x] Robertson (1994a: footnote 1) notes that the only reason they didn't do worse at the first TREC was that there was some old code that was accidentally truncating documents.

[xi] BM25 is also used in commercial products and web search engines: *"BM25 is used as one of the most important signals in large web search engines, certainly in Microsoft's Bing, and probably in other web search engines too. "* http://irsg.bcs.org/informer/2013/10/celebrating-stephen-robertson%E2%80...

[xii] Robertson also justified using the mixture model by suggesting the following:

"A simple normalistion (dividing TF by DL) would have the effect of giving the same score to a document of length DL in which a term occurs TF times, as to a document of length 2DL in which the same term occurs 2TF times. But the crudity of the assumption is likely to lead to a bias in the above normalization. That is, the 2DL document is unlikely to require a ***smaller*** *score than the DL document , and it may be justifieable to give it a larger one (e.g. if wordiness suggests greater elaboration rather than just repetition). The slightly more complex normalization suggested below, a mixture of no normalization at all and the above simple normalization allows for this"*

Sparck Jones et al. (2000 p 813)

[xiii] The Inquery system from UMASS CIIR also was doing significantly better than SMART (Singhal 1996b:619)

[xiv] See for example: Singhal et al. (1996b. p 626) "...the probability of retrieving a document of a certain length using the weighting scheme of the Okapi system has a very stong correlation with the probability of finding a relevant document of about the same length...We believe that this strong correlation is the main reason behind the superior retrieval effectiveness of Okapi's term weighting scheme."

[xv] Note that the probability of relevance increases with document length. Losada et al. (2008) cite other work showing that the probability of relevance increases with document length. However, they go on to investigate several collections and compare document lengths in the collection as a whole with the document lengths of the judged documents. They conclude that the prior findings of a relationship between document length and relevance may be a result of pooling bias.

[xvi] Singhal 1996b p 627-628.

[xvii] Buckley (in Voorhees and Harman 2005: p 308) reports that "pivoted document normalization *"..fundamentally changed every approach we had been taking"* He then describes how the SMART team discovered that many of the techniques they used to improve retrieval in TREC-3 were successful only because they incidentally made some change to the document length of retrieved documents.

[xviii] Both DFR and IB require both choosing a normalization algorithm from a list of available algorithms and tuning a length normalization parameter: "c" (See http://lucene.apache.org/core/4_9_0/core/org/apache/lucene/search/simila...). DFR does have several parameterless models. One of them is DFRFree (http://terrier.org/docs/current/javadoc/org/terrier/matching/models/DFRe...). The language modeling algorithms also have tuning parameters that are called "smoothing" parameters, but Losada and Azzopardi (2008) and Zhai and Lafferty (2001) find that the smoothing parameters also serve a length normalization function. There is a substantial amount of research to that attempts to create models that do not need parameters and can still be computed efficiently, however, to our knowledge, these methods are not widely used. See for example Amati 2006, He and Ounis 2007, Lv and Zhai 2011.

[xix] See Robertson and Zaragoza 2009 section 5 for discussion of parameter optimization. A recent paper (Schuth et al. 2014) suggests that parameters (of BM25) can be learned using click logs, which would avoid the need to build a test collection and gather manual relevance judgments. However, the paper used a simulation of user clicks based on query logs and judgments from a test collection to test the method. Hopefully the method will be implemented and tested in a live search engine which would make the results more convincing. One problem with any click-based tuning method is that such methods are good for fine-tuning an algorithm that is already good, but not very good for algorithms that may place relevant documents below the 2nd or 3rd page of results. Users are unlikely to visit results pages beyond the first few pages, so the learning algorithm is unlikely to get clicks/feedback about relevant documents far down on the result list.

[xx] Zobel and Moffatt (1998) found it was difficult to predict which types of parameter combinations would work best for a particular combination of query type and collection. (See also endnote iv).

[xxi] In discussion of a precursor to BM25, Robertson states: "The drawback of these two models is that the theory says nothing about the estimation of the constants or rather parameters k1 and k2. It may be assumed that these depend on the database and probably also on the queries and on the amount of relevance information available." 1993 sec 7.1 In an article on the history of IR evaluation Robertson states "...theories or models tend to be the subject of experimental investigation *only* in terms of the effectiveness of the resulting system. Seen as an application of the usual scientific method, of challenging theories by trying to derive falsifiable consequnces, which may then be tested experimentally, this is extremely limited. (Robertson 2008 p 452) See also the discussion in Fuhr 2012 and Fuhr 2010. The REFORM project led by ChengXiang Zhai tries to address these issues:

"Although many different retrieval models have been proposed and studied ever since the beginning of the field of IR, there has been no single model that has proven to be the best. Theoretically well-motivated models all need heuristic modifications to perform well empirically. It has been a long-standing scientific challenge to develop principled retrieval models that also perform well empirically. Existing retrieval models have several fundamental limitations:

(1) The performance of a retrieval model is highly sensitive to the document collections and queries in an unpredictable way.

(2) A model that performs well on some data set may perform poorly on another data set.

(3) Heavy parameter tuning must be done manually to achieve optimal performance.

In this project, we aim to develop novel retrieval models that are robust (w.r.t. the variation of document collections and queries), effective (in terms of retrieval accuracy), and can guarantee optimality to certain extent." http://sifaka.cs.uiuc.edu/ir/proj/reform/

[xxii] Another option used in research situations when there is no production collection is to split a test collection into a test and training collection. The algorithms are tuned on the training collection and then tested on the test collection. In the TREC ad hoc runs for example, generally training data from previous TREC runs is used (See for example Singhal 1996b.)

[xxiii] Jones et al. 2014 find that experiments with breaking a collection into "subcollections" raise several questions about what constitutes a "representative" collection. When HathiTrust moves to an algorithm

that requires tuning, we would either need to create a test collection or use an appropriate existing test collection for tuning. The INEX Book Track has a collection of 50,000 books that is somewhat similar to the content of HathiTrust. There are a number of issues with using this collection. The HathiTrust corpus contains over 11 million volumes or more than two orders of magnitude more documents. With the exception of some work on why link evidence and anchor text (used in production scale web search engines) did not appear to help the ad hoc task in the web (See Koolan and Kamps 2010 and references therin), we are not aware of any research on how the findings of experiments in relatively small test collections can be extrapolated to production collections that are two or more orders of magnitude greater in size. The queries used in the 2007 INEX Book Track are in general much more underspecified than most of the queries we see in the HathiTrust query logs. The INEX Book Track collection and the issues of building a test collection will be discussed in future blog posts.

[xxiv] Collecting a sufficient quantity of high quality relevance judgments is the major bottleneck in creating test collections for information retrieval research. There is a large literature on the trade-offs involved in various methods for reducing the cost of gathering judgments. See Sanderson (2010) and Carterette et al. (2010) for an overview of the issues. Kazai (2009) details some of the issues in creating the INEX book track collections. Kazai's work include extensive investigation of crowdsourcing relevance judgements for book retrieval. See publications listed at http://www.gabriella-kazai.com. Alonso (2013) provides a good discussion on the amount of work involved in doing industrial strength crowdsourcing for relevance judgment. A future blog post will discuss test collections in more detail.

Printer-friendly version

Add new comment

Your name

E-mail

The content of this field is kept private and will not be shown publicly.

Homepage

Subject

Comment *

CAPTCHA

This question is for testing whether you are a human visitor and to prevent automated spam submissions.

I'm not a robot

reCAPTCHA
Privacy - Terms

Save Preview