Where to stop reading a ranked list?

Arampatzis, A.; Nussbaum, N.; Kamps, J.

Publication date
2008

Published in
The seventeenth Text REtrieval Conference (TREC 2008) notebook

Where to Stop Reading a Ranked List? *

Avi Arampatzis[1] **Nir Nussbaum**[2] **Jaap Kamps**[1,2]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] ISLA, Informatics Institute, University of Amsterdam

1 Introduction

In recall-oriented retrieval setups, such as the Legal Track, ranked retrieval has a particular disadvantage in comparison with traditional Boolean retrieval: there is no clear cut-off point where to stop consulting results. It is expensive to give a ranked list with too many results to litigation support professionals paid by the hour. This may be one of the reasons why ranked retrieval has been adopted very slowly in professional legal search.[1]

The "missing" cut-off remains unnoticed by standard evaluation measures: there is no penalty and only possible gain for padding a run with further results. The TREC 2008 Legal Track addresses this head-on by requiring participants to submit such a cut-off value (K for relevant and K_h for highly relevant results) per topic where precision and recall are best balanced. This year we focused solely on selecting K for optimizing the given F_1-measure. We believe that this will have the biggest impact on this year's comparative evaluation.

The rest of this paper is organized as follows. In Section 2 we describe the experimental set-up. The method for determining K is presented in Section 3. It depends on the underlying score distributions of relevant and non-relevant documents, on which we elaborate in Section 4. In Section 5 we discuss our official submissions, results, and additional experiments. Finally, we summarize the findings in Section 6.

2 Experimental Set-up

We employed the same experimental set-up as last year, fully described in [4]. Specifically, document pre-processing, indexing, and retrieval model, are the same as for last year's post-submission run tagged in the last-cited study as **text-only**, i.e. our best run in terms of mean average precision.

In pre-processing this year's topics, we used the `RequestedText` field stop-listed by an extended list in which we manually included low-content words based on the topics of 2006 and 2007.

More information about the collection, topics, and evaluation measures can be found at the TREC Legal web-site.

*This is the TREC Notebook version of this paper. It may be inaccurate or contain errors. For the final version, check the TREC 2008 Proceedings.

[1] In fact, to the surprise of many, at the TREC 2007 Legal Track the Boolean reference run outperformed the ranked retrieval models at the rank cut-off of the Boolean set size.

3 Thresholding a Ranked List

Essentially, the task of selecting K is equivalent to thresholding in binary classification or filtering. Thus, we recruited a method first appeared in the TREC 2000 Filtering Track, namely, the *score-distributional threshold optimization* (s-d) [2, 3]. The method goes as follows.

3.1 The S-D Threshold Optimization

Let us assume an item collection of size n, and a query for which all items are scored and ranked against. Let $P(s|1)$ and $P(s|0)$ be the probability densities of relevant and non-relevant documents as a function of the score s, and $F(s|1)$ and $F(s|0)$ their corresponding *cumulative distribution functions* (cdfs). Let $G_n \in [0,1]$ be the fraction of relevant documents in the collection, also known as *generality*.

The total number of relevant documents in the collection is given by

$$R = n\,G_n \tag{1}$$

while the numbers of relevant and non-relevant documents with scores $> s$ are

$$R_+(s) = R\,(1 - F(s|1)) \tag{2}$$

$$N_+(s) = (n - R)\,(1 - F(s|0)) \tag{3}$$

respectively. The numbers of the remaining relevant and non-relevant documents with scores $\leq s$ respectively are

$$R_-(s) = R - R_+(s) \tag{4}$$

$$N_-(s) = (n - R) - N_+(s)\,. \tag{5}$$

In this way, any effectiveness measure M based on the above four document counts can be calculated at any score or rank. Assuming that the larger the M the better the effectiveness, the optimal rank is

$$K = \arg\max_k \{M\,(R_+(s_k), N_+(s_k), R_-(s_k), N_-(s_k))\}$$

where s_k is the score of the kth ranked document.[2] The only unknown quantities which have to be estimated are the densities $P(s|1)$, $P(s|0)$, and the generality G_n.

[2] Strictly speaking, if the expression is maximized at k but $M(s_k^-) < M(s_k)$, then $K = k - 1$. This allows for K to become 0, meaning that no document should be retrieved.

Given these, we have so far a clear theoretical answer to predicting K, R, and even real probabilities of relevance as we will see next.

3.2 Probability Thresholds

Given the two densities and the generality defined earlier, scores can be normalized to probabilities of relevance straightforwardly [2, 11] by using the Bayes' rule.

Normalizing to probabilities is very important in tasks where several rankings need to be fused or merged such as in meta-search/fusion or distributed retrieval. This may also be important for thresholding when documents arrive one by one and decisions have to be made on the spot, depending on the measure under optimization. Nevertheless, it is unnecessary for thresholding rankings since optimal thresholds can be found on their scores directly, and it is furthermore unsuitable given F_1 as the evaluation measure.

While for some measures there exists an optimal *fixed* probability threshold, for others it does not. D. Lewis formulates this in terms of whether or not a measure satisfies the *probability thresholding principle*, and proves that the F measure does not satisfy it [10] . In other words, how a system should treat documents with, e.g., 50% chance of being relevant depends on how many documents with higher probabilities are available. Consequently, for such measures, what we should be looking for is a different *score* or *rank* threshold for each ranking.

3.3 Beyond Binary Relevance

The s-d thresholding assumes binary relevance. Thus, this year's three-way classification into non-relevant, relevant, and highly relevant cannot be dealt with—in a theoretically sound way—in the context of s-d.

In order to set a rank-threshold for the highly relevant, we try an *ad-hoc* approach. We just set K_{hz} at the rank with a corresponding score nearest to $\mathrm{E}(Z) + z\sqrt{\mathrm{Var(Z)}}$, $z \in \mathbb{N}^+$, where $\mathrm{E}(Z)$ and $\mathrm{Var}(Z)$ the expectation and variance respectively of a random variable Z distributed as $P(s|1)$. We cap K_{hz} at K. Obviously, simply because $P(s|0)$ does not contribute in any way, the method does not optimize F_1.

4 Score Distributions

Let us know elaborate on the form of the two densities $P(s|1)$ and $P(s|0)$ of Section 3.1 and their estimation.

Score distributions have been modeled since the early years of IR with various known distributions [5, 6, 15, 16]. However, the trend during the last few years, which has started in [3] and followed up in [1, 2, 7, 11, 17], has been to model score distributions by a mixture of normal-exponential densities: normal for relevant, exponential for non-relevant.

Despite its popularity, it was pointed out recently that, under a hypothesis of how systems should score and rank documents, this particular mixture of normal-exponential presents a theoretical anomaly [13]. In practice, nevertheless, it has stand the test of time in the light of

- its (relative) ease to calculate,
- good experimental results, and
- lack of a proven alternative.

In this paper, we do not set out to investigate alternative mixtures. We just refine the model to account for practical situations, for its theoretical anomaly, and improve the computation methods. We also check its goodness-of-fit to empirical data using a statistical test; a check that has not been done before as far as we are concerned. At the same time, we explicitly state all parameters involved, try to minimize their number, and find for them a robust set of values.

4.1 Estimating Score Densities

The normal-exponential mixture has worked best under the availability of some relevance judgments which serve as an indication about the form of the component densities [3, 7, 17]. In filtering or classification, usually some training data—although often biased—are available. In the current task, however, no relevance information is available.

A method was introduced in the context of fusion which recovers the component densities without any relevance judgments using the Expectation Maximization (EM) algorithm [11]. In order to deal with the biased training data in filtering, the EM method was also later adapted and applied for thresholding tasks [1].[3] Nevertheless, EM was found to be "messy" and sensitive to its initial parameter settings [1, 11].

4.2 Recovering the Mixture

In the context of s-d, the total score distribution is written as

$$P(s) = (1 - G_n)\, P(s|0) + G_n P(s|1) \qquad (6)$$

where $G_n \in [0, 1]$.

Let us assume that a retrieval model produces, in theory, scores in $[s_{\min}, s_{\max}]$, where $s_{\min} \in \mathbb{R} \cup \{-\infty\}$ and $s_{\max} \in \mathbb{R} \cup \{+\infty\}$. By using an exponential distribution, which has semi-infinite support, the applicability of the s-d model is restricted to those retrieval models for which $s_{\min} \in \mathbb{R}$. Consequently, the two densities are given by

$$P(s|0) = \lambda \exp\left(\lambda\left(s_{\min} - s\right)\right), \quad \lambda > 0\,,\ s \geq s_{\min} \qquad (7)$$

$$P(s|1) = \frac{1}{\sigma}\,\varphi\left(\frac{s-\mu}{\sigma}\right), \quad \sigma > 0\,,\ \mu, s \in \mathbb{R} \qquad (8)$$

and $\varphi(.)$ is the density function of the standard normal distribution, i.e. with a mean of 0 and standard deviation of 1. Hence, there are 4 parameters to estimate, λ, μ, σ, and G_n, which we do with EM.

Note that although we already generalized somewhat here by introducing a *shifted exponential*, the mix has always had

[3] Another method for producing unbiased estimators in filtering can be found in [17], but it requires relevance judgements.

a support incompatibility problem; while the exponential is defined at or above some $s_{\min}$, the normal has a full real axis support. First we deal with this issue.

4.2.1 Down-truncated Rankings

For practical reasons, rankings are usually truncated at some rank $t \leq n$. Even what is usually considered a full ranking is in fact a collection's subset of the non-zero scored documents. In TREC Legal 2007 and 2008, t was $25,000$ and $100,000$ respectively. This may result to a left-truncation of $P(s|1)$ which at least in the case of the 2007 data is significant. For 2007 it was estimated that there were more than $25,000$ relevant documents for 13 of the 43 Ad Hoc topics (to a high of more than $77,000$) and the median system was still achieving 0.1 precision at ranks of $20,000$ to $25,000$.

In order to take into account a possible truncation of $P(s|1)$, we use a *left-truncated normal distribution* instead. If the truncation score is s_t, Equation 8 is replaced by

$$P(s|1) = \frac{\frac{1}{\sigma}\varphi\left(\frac{s-\mu}{\sigma}\right)}{1-\Phi\left(\alpha\right)}\,, \quad \alpha = \frac{s_t-\mu}{\sigma} \tag{9}$$

where $\Phi(.)$ is the cumulative distribution function of $\varphi(.)$. For $P(s|0)$, we simply shift the exponential (Equation 7) by s_t instead of $s_{\min}$.

If G_t is the fraction of relevant documents in the truncated ranking, the Equations 1 to 3 must be re-written as

$$R = t\,G_t\left(1+\frac{\Phi\left(\alpha\right)}{1-\Phi\left(\alpha\right)}\right) = \frac{t\,G_t}{1-\Phi(\alpha)} \tag{10}$$

$$R_+(s) = t\,G_t\,\left(1-F(s|1)\right) \tag{11}$$

$$N_+(s) = t\,\left(1-G_t\right)\left(1-F(s|0)\right) \tag{12}$$

while Equations 4 and 5 remain the same. $F(s|1)$ is now the cdf of the truncated normal.[4]

Note that for $s_t \ll \mu$, $\Phi\left(\alpha\right) \approx 0$ and $P(s|1)$ becomes a full non-truncated normal. Consequently, using a truncated normal is a valid choice even when rankings are not truncated. It is also valid when rankings are truncated but no relevant documents are removed by the truncation. This improvement makes the model more general, and it indeed produces better fits on our data. [5]

[4] We do not give here formulas or computation methods for the cdfs due to space limitations; these can easily be found in relevant literature.

[5] With this modification, and setting $s_t = s_{\min}$ and $G_t = G_n$, we have reached a new mixture model for score distributions, i.e. *truncated normal-exponential*, with a semi-infinite support in $[s_{\min}, +\infty)$, $s_{\min} \in \mathbb{R}$. The underlying density of relevant documents has a discontinuity at $s_{\min}$: $R_+(s_{\min}) < R$, but $R_+(s_{\min}^-) = R$.

In practice, scores may be naturally bounded (by the retrieval model) or truncated to the upside as well. For example, cosine similarity scores are naturally bounded at 1. Scores from probabilistic models with a (theoretical) support in $(-\infty, +\infty)$ are usually mapped to the bounded $(0, 1)$ via a logistic function. Other retrieval models may just truncate at some maximum number for practical reasons.

In the case of probabilistic models, scores can be "unfolded" again with a logit(.) transformation; in principle, the same can be done for any retrieval model with scores in $[0, 1]$, with a special treatment of 0 and 1. Whether the normal-exponential mixture fits the "unfolded" data better or worse is an open question. In any case, using any distribution with a right tail going to infinity for modeling scores would have the same theoretical issue; practically, it seems of no significance.

4.2.2 Score Preprocessing

Beyond using all scores available and in order to speed up the calculations, we also tried to uniformly down-sample the data with probabilities of 0.1 and 0.5.[6]

Our scores have a resolution of 10^{-6}. Obviously, LUCENE rounds or truncates the output scores, destroying information. In order to smooth out the effect of rounding in the data, we add $\Delta s = \mathrm{rand}(10^{-6}) - 0.5 * 10^{-6}$ to each datum point, where $\mathrm{rand}(x)$ returns a uniformly-distributed real random number in $[0, x)$.

In order to obtain better exponential fits we may left-truncate further than a possibly already existing truncation. We bin the scores (as described in Section 4.2.4), find the bin with the most scores, and if that is not the first bin then and remove all scores in previous bins.

4.2.3 Expectation Maximization

EM is an iterative procedure which converges locally [8]. Finding a global fit depends largely on the initial settings of the parameters.

Initialization We tried numerous initial settings, but no setting seemed universal. While some settings helped a lot some fits, they had a negative impact on others. Without any indication of the form, location, and weighting of the component densities, the best fits overall were obtained for randomized initial values, preserving also the generality of the approach:[7]

$$G_{t,\mathrm{init}} = \mathrm{rand}(1)\,, \quad \lambda_{\mathrm{init}} = \left(\epsilon + \mathrm{rand}(\mu_s - s_t)\right)^{-1}$$

$$\mu_{\mathrm{init}} = s_{\min} + \mathrm{rand}(s_1 - s_{\min})$$

$$\sigma^2_{\mathrm{init}} = \left(1 + c_1 \mathrm{rand}\left(1\right)\right)^2 \max\left(\epsilon^2,\, \sigma_s^2 - \lambda_{\mathrm{init}}^{-2}\right)$$

where s_1 is the maximum score datum, μ_s and σ_s^2 are respectively the mean and variance of the score data, ϵ is an arbitrary small constant which we set equal to the width of the bins (see Section 4.2.4), and $c_1 \in [0, +\infty)$ is another constant which we explain below.

The μ_{init} given is suitable for a full normal, and its range should be expanded for a truncated one because the mean of the corresponding full normal can be way below $s_{\min}$. Further, μ_{init} can be restricted based on the hypothesis that for

[6] In order not to complicate things, we do not include the down-sampling into the formulas in this paper; it is not difficult to see where things should be weighted inversely proportional to the sampling probability.

[7] With some (even biased) training data, suitable initial parameter settings are given in [1]. Without any training data, assuming that the relevant documents are much fewer than non-relevant by rank t, initial parameters can be estimated as described in [11]; unfortunately this assumption cannot be made in TREC Legal due to topics with very large estimated numbers of relevant documents.

good systems the expected relevant score[8] should be greater than the expected value of the exponential, which (accounting for the shift) is $\lambda_{\text{init}}^{-1} + s_t$. We have not yet worked out these improvements.

The variance of the initial exponential is $\lambda_{\text{init}}^{-2}$. Assuming that the random variables corresponding to the normal and exponential are uncorrelated, the variance of the normal is $\sigma_s^2 - \lambda_{\text{init}}^{-2}$ which, depending on how λ is initialized, could take values ≤ 0. To avoid this, we take the max with the constant. For a full normal, $c_1 = 0$, while for a truncated one, $c_1 > 0$, because the variance of the corresponding full normal is larger than what is observed in the truncated data. We set $c_1 = 0.25$.

Update Equations For $t \leq n$ observed scores $s_1 \dots s_t$, and a full normal density $P(s|1)$ (Equation 8), the update equations are

$$G_{t,\text{new}} = \frac{\sum_i P_{\text{old}}(1|s_i)}{t} \quad \lambda_{\text{new}} = \frac{\sum_i P_{\text{old}}(0|s_i)}{\sum_i P_{\text{old}}(0|s_i)(s_i - s_t)}$$

$$\mu_{\text{new}} = \frac{\sum_i P_{\text{old}}(1|s_i)s_i}{\sum_i P_{\text{old}}(1|s_i)} \quad \sigma^2_{\text{new}} = \frac{\sum_i P_{\text{old}}(1|s_i)(s_i - \mu_{\text{new}})^2}{\sum_i P_{\text{old}}(1|s_i)}$$

$P(j|s)$ is given by Bayes' rule $P(j|s) = P(s|j)P(j)/P(s)$, $P(1) = G_t$, $P(0) = 1 - G_t$, and $P(s)$ by Equation 6.

Correcting for Truncation The above update equations do not hold for a truncated normal density $P(s|1)$ (Equation 9), because the calculated μ_{new} and σ^2_{new} at each iteration are the expected value and variance respectively of the *truncated* distribution, not the mean μ and variance σ^2 of the corresponding non-truncated one. Instead of looking for new EM equations, we rather correct to the right values using a simple approximation.

We note that for a normally-distributed random variable X with μ and σ^2 and a left-truncation at s_t, the expected value and variance are

$$\text{E}(X|s_t < X) = \mu + \sigma\,\psi(\alpha)\,, \quad \psi(\alpha) = \frac{\varphi(\alpha)}{1 - \Phi(\alpha)} \tag{13}$$

$$\text{Var}(X|s_t < X) = \sigma^2\left[1 - \psi(\alpha)\left(\psi(\alpha) - \alpha\right)\right]$$

where α is given by Equation 9 and contains the unknown μ and σ^2. We approximate α with

$$\alpha' = \frac{s_t - \mu_{\text{old}}}{\sqrt{\sigma^2_{\text{old}}}}$$

using the values from the previous iteration, and at the end of the current iteration we correct the calculated μ_{new} and σ^2_{new} as

$$\mu_{\text{new}} \leftarrow \mu_{\text{new}} - \sigma_{\text{old}}\,\psi(\alpha') \tag{14}$$

$$\sigma^2_{\text{new}} \leftarrow \sigma^2_{\text{new}}\left[1 - \psi(\alpha')\left(\psi(\alpha') - \alpha'\right)\right]^{-1} \tag{15}$$

This simple approximation works, but sometimes it seems to increase the number of iterations needed for convergence, depending on the accuracy targeted. The accuracy and number of iterations issue will be discussed later.

4.2.4 Chi-Square Goodness of Fit

To check the quality of the fits, we bin the scores and calculate the χ^2 statistic

$$\chi^2 = \sum_i \frac{|O_i - E_i|^2}{E_i} \tag{16}$$

where O_i and E_i are the observed and expected frequencies respectively for bin i [12]. The expected frequency is calculated by

$$E_i = t\left(F(s_{i,a}) - F(s_{i,b})\right)$$

where $s_{i,a}$ and $s_{i,b}$ are respectively the lower and upper score limits of bin i, and $F(s) = (1 - G_t)F(s|0) + G_t F(s|1)$ is the cumulative distribution function of the mixture.

For the χ^2 approximation to be valid, E_i should be at least 5, thus we may combine bins in the right tail when $E_i < 5$. When the last E_i does not reach 5 even for $b = +\infty$, we only then apply the Yates' correction, i.e. subtract 0.5 from the absolute difference of the frequencies in Equation 16 before squaring. The χ^2 statistic is sensitive to the choice of bins.

Score Binning For binning, we use the optimal number of bins as this is given by the method described in [9]. The method considers the histogram to be a piecewise-constant model of the underlying probability density. Then, it computes the posterior probability of the number of bins for a given data set. This enables one to objectively select an optimal piecewise-constant model describing the density function from which the data were sampled.

4.2.5 Rejecting Fits on IR Grounds

Some fits, irrespective of their quality, can be rejected on IR grounds. Firstly, it should hold that $N + R = n$, where N is the estimated number of non-relevant items including the ones below s_t (in parallel to Equation 10):

$$N = \frac{t\,(1 - G_t)}{1 - F(s_t - s_{\text{min}}|0)}$$

The problem is that the exponential is usually not a good fit below some score, making the N estimate inaccurate.[9] Nevertheless, the model should not be far off even in the whole distribution. Allowing for some over- and under-estimation, we settle for

$$\frac{|n - (N + R)|}{N + R} < c_2 \in (0, +\infty) \tag{17}$$

[8] Note that the expected value of the relevant score is neither μ nor $\text{E}(X|s_t < X)$ or $\text{E}(X|s_{\text{min}} < X)$ (see Equation 13). It is the expected value of the random variable Z distributed as a mixture of a left-truncated normal at s_{min} and a peak at s_{min} representing the not-retrieved-at-all relevant items. Assuming that the lower truncation at s_{min} is insignificant, E(Z) can be approximated by μ.

[9] This is also the reason why we estimate $N_+(.)$ and $N_-(.)$ as a fraction or part of $(n - R)$ respectively and not out of N directly.

and set $c_2 = 20$. If it does not hold, we reject the fit. This should handle a few extremities.

Secondly, concerning the two underlying random variables Y and Z, one would expect

$$\frac{1}{\lambda} + s_t = \mathrm{E}(Y) \leq E(Z) \approx \mu \tag{18}$$

This is rather only a hypothesis—not a requirement—that good systems should satisfy and there are no guarantees. We reject fits that do not satisfy it.

4.2.6 Putting It All Together

Parameter Estimation with EM We initialize EM with random parameter values as described above, and iterate the update equations until the absolute differences between the old and new values for μ, λ^{-1}, and $\sqrt{\sigma}$ are all less than $.001\,(s_1 - s_{\min})$, and $|G_{t,\text{new}} - G_{t,\text{old}}| < .001$. Like this we target an accuracy of 0.1% for scores and 1 in a 1,000 for documents.

Rarely, and for high accuracies only, the approximation we do in Equations 14 and 15 possibly handicaps EM convergence; the intended accuracy is not reached for up to 1,000 iterations. Generally, convergence happens in 10 to 50 iterations depending on the number of scores (more data, slower convergence), and even with the approximation EM produces considerably better fits than when using a non-truncated normal. To avoid getting locked in a non-converging loop, despite its rarity, we cap the number of iterations to 100.

After EM stops, we calculate the χ^2 of the fit with the binned score data. Since we estimate 4 parameters, the degrees of freedom of the χ^2 distribution is $M-4-1$, where M is the number of bins. If the χ^2 of the fit is below the critical value of the corresponding χ^2 distribution at a significance level of 0.05, we accept the fit. If not, we randomize the initial values and repeat EM for up to 100 times or until a fit passes the test. If none passes the test, we keep the best one. We run EM at least 10 times, even if we get a pass earlier. Perhaps a maximum of 100 EM runs is an overkill, but we found that convergence to a global optimum is very sensitive to initial conditions.

Different EM fits on the same data can result to slightly different degrees of freedom due to combining bins as in Section 4.2.4. To keep track of the best fit found, we compare the quality of fits with their corresponding χ^2 upper-probability. The higher the probability, the better the fit. We initialize the χ^2 upper-probability at its value of an exponential-only fit, by setting $\lambda = 1/(\mu_s - s_t)$.

Quality of the Fits and Sampling Concerning the resulting fits after the above procedure, at a significance level of 0.05, all fits but 2 or 3 on the Legal 2007 data get a χ^2 larger than the critical value and they can be rejected. Nevertheless, all look reasonably well to the eye. The number of scores and bins seem to play a big role in the quality of the fits and the χ^2 test.

Down-sampling has the effect of eliminating some of the right tails, leading to fewer bins when binning the data. The fewer the scores, the less EM runs are needed for a good fit. Down-sampling the scores helps the χ^2 test. At around 1,000 to 10,000 scores, almost all fits pass the test.

4.3 The Recall-Fallout Convexity Hypothesis

From the point of view of how scores or rankings[10] of IR systems should be, S. Robertson formulates the recall-fallout convexity hypothesis [13]:

> *For all good systems, the recall-fallout curve (as seen from [. . .] recall=1, fallout=0) is convex.*

Similar hypotheses can be formulated as a conditions on other measures, e.g., the probability of relevance should be monotonically increasing with the score; the same should hold for *smoothed* precision. Although, in reality, these conditions may not always be satisfied, they are expected to hold for good systems because their failure implies that systems can be easily improved.

As an example, let us consider smoothed precision. If it declines as score increases for a part of the score range, that part of the ranking can be improved by a simple random reordering [14]. This is equivalent of "forcing" the two underlying distributions to be uniform (i.e. have linearly increasing cdfs) in that score range. This will replace the offending part of the precision curve with a flat one—the least that can be done— improving the effectiveness of the system.

Such hypotheses put restrictions on the relative forms of the two underlying distributions. The normal-exponential mixture violates such conditions, only (and always) at both ends of the score range. Although the low-end scores are of insignificant importance, the top of the ranking is very significant, especially for low R topics. The problem is a manifestation of the fact that an exponential tail extends further than a normal one.

To complicate matters further, our data suggest that such conditions are violated at a different score s_c for the probability of relevance and for precision. Since the F-measure we are interested in is a combination of recall and precision (and recall by definition cannot have a similar problem), we find s_c for precision. We "fix"' the distributions only when $s_c < s_1$; otherwise, the theoretical anomaly does not affect the score range which is mostly the case.

If $s_{\max}$ is finite (which theoretically does not agree with the $[s_{\min}, +\infty)$ support of the improved version of the mixture we present in this paper), then two uniform distributions can be used in $[s_c, s_{\max}]$ as mentioned earlier. Alternatively, preserving the theoretical support, the relevant documents distribution can be forced to an exponential in $[s_c, +\infty)$ with the same λ as this of the non-relevant. We apply the alternative.

[10] Score or rank can be used interchangeably in this Section.

Table 1: Ranking quality for the Legal 2008. The highest, lowest, and median are of the 23 submissions using the `RequestText` field.

Run	Relevant			Highly Relevant		
	$Prec@5$	$Recall@B$	$F_1@R$	$Prec@5$	$Recall@B$	$F_1@R_h$
uva-base	0.5000	0.2030	0.1729	0.2583	0.4541	0.1302
uva-x$*$	0.4846	0.2036	0.1709	0.2500	0.4582	0.1474
highest	0.5923	0.2779	0.2173	0.3250	0.5001	0.1770
median	0.4154	0.2036	0.1709	0.1917	0.3506	0.1152
lowest	0.0538	0.0729	0.0694	0.0000	0.0902	0.0299

5 Experiments

Next we describe our runs, present and discuss the results.

5.1 Official and Additional Runs

We submitted 4 official runs. The first 3 are based on the same ranking, with various methods of selecting thresholds:

uva-xconst K was set to 16,904, i.e. the average R in 2007, for all topics, and $K_h = 0.5K = 8,452$.

uva-xb $K = \bar{R}(B/\bar{B})$ per topic and $K_h = 0.5K$, where B is the number of results of the negotiated Boolean query, $\bar{B}$ the mean B value, and $\bar{R}$ the average R in 2007.

uva-xk K and K_h were determined by the s-d method described in Section 3.1 and computed, more or less, as described in Section 4.

uva-base K and K_h were determined by the s-d method as in **uva-xk**. The standard SMART stop-list was used instead of the extended one.

In fact, for the official runs we used an earlier rougher version of the computations we give in Section 4, so what is presented in this paper is the improved method.

The improvements that have the greatest impact on end-user effectiveness are:

1. Replacement of the normal with a truncated normal distribution to account for possibly missing relevant documents from the truncated rankings.
2. EM is run with different initial parameters, and better termination methods. We also now run it up to 100 times instead of 10.
3. We used the square error before to select the best fit; we replaced this with the χ^2 which is more suitable for distributions.
4. Optimal binning. Before we used a fixed number of $\max(5, t/200)$ bins, which gave 500 bins (or a bit less after a left-truncation of the data) for the 2008 rankings.
5. Correcting for precision monotonicity; we were correcting the probability of relevance monotonicity before, not knowing that the offending points may differ.

Consequently, we provide here an additional run. We also made runs with K set to B, or K set to 2008's mean R:

uva-xk2 Based on the same ranking as **uva-xk**, K and K_h are computed exactly as described in Section 4.

uva-xb2 K is set to B, the size of the result set of the Boolean query.

uva-xconst2 K is set to 82,403, the mean estimated R in 2008, and K_h to the smallest integer larger than $K/2$.

All 2008 rankings are truncated at 100k items.

5.2 Results and Discussion

We first discuss the overall quality of the rankings, and then the main topic of this paper—estimating the cut-off K.

The top half of Table 1 shows several measures on the two underlying rankings, **uva-base** and **uva-x**$*$. We show precision at 5 (all top-5 results were judged by TREC); estimated recall at B; and the F_1 of estimated precision and estimated recall at R (i.e. the estimated number of relevant documents). As expected, the two rankings perform similarly.

To determine the quality of our rankings in comparison to other systems, we show the highest, lowest, and median performance of all submissions in the bottom half of Table 1. As it turns out, **uva-x**$*$ obtains exactly the median performance for $Recall@B$ and $F_1@R$ when using all relevant documents in evaluation. Both rankings fare somewhat better at $Prec@5$ and in evaluating with the highly relevant documents only. It is clear that our rankings are far from optimal in comparison with the other submissions. On the negative side, this may affect the performance of the s-d method, since the retrieval scores are its only input. On the plus side, it makes our rankings good representatives of the median-quality ranking.

Table 2 shows the results for the various thresholding methods. Looking at the four official runs, we see that the (rough) s-d method (**k**) stays behind choosing a constant threshold based on the estimated number of relevant documents in 2007 (**const**) and thresholding at the B values (**b**). We see the same pattern for highly relevant documents. Although the comparison is in some sense unfair, the mean estimated number of relevant items is generally not known and the B values are based on a negotiated query, we expected the s-d method to do better. Indeed, the improved version

Table 2: End-results for the Legal 2008. The highest, lowest, and median are of the 23 submissions using the `RequestText` field.

Run	Relevant $F_1@K$	Highly Rel. $F_1@K_h$
const (2007 mean R)	0.1264	0.0728
b (modified B)	0.1030	0.0661
k	0.0689	0.0600
k (uva-base ranking)	0.0650	0.0523
const2 (2008 mean R)	0.1566	0.0813
b2 (standard B)	0.1416	0.0924
k-improved with K_{h1}	0.1374	0.0683
k-improved with K_{h2}	0.1374	0.0570
highest	0.1848	0.0882
median	0.0974	0.0481
lowest	0.0051	0.0003

of the s-d method (**k-improved**) leads to substantially better results, well above constant thresholding (**const**).

We also show the highest, lowest, and median performance over the 23 submissions. Note that the actual value of $F_1@K$ is a result of both the quality of the underlying ranking *and* choosing the right threshold. As seen earlier, our ranking has the median $Recall@B$ and $F_1@R$. With the estimated threshold of the s-d model, the $F_1@K$ is 0.1374, well above the median score of 0.0974.

There is still amble room for improvement. The $F_1@B$ is higher with 0.1416, and the $F_1@R$ is 0.1709. Also updating the constant threshold to the mean number of estimated R for 2008 (**const2**) does lead to a better score. We have also calculated the optimal threshold, and $F_1@K_{\text{opt}}$ is 0.2325. Obviously, R or K_{opt} are not known in a operational system, so $F_1@R$ and $F_1@K_{\text{opt}}$ serve us here as "ceilings" of performance, with the latter being the best possible.

6 Conclusions

In this paper, we studied the problem of finding an "optimal" point to stop reading a ranked list, by selecting thresholds that optimize the given F_1-measure. Our approach uses the score-distributional threshold optimization (s-d), a technique proven effective for binary classification in earlier years. We use no other input than the document scores of a standard retrieval run, fit a mixture of normal-exponential distributions (normal for relevant, and exponential for non-relevant documents), and calculate the optimal score threshold given the estimated distributions and their contributing weight.

The experiments confirm that the s-d method is effective for determining the thresholds, although there is still clear room for improvement: the effectiveness varies considerably per topic (with an average of 60% of the optimal F_1). Assuming that a normal-exponential mixture is a good approximation for score distributions and that no relevance information is available, we believe that the estimation method described in this paper is *a)* as general as possible, *b)* it deals with all known theoretical "anomalies" and practical difficulties, and consequently, *c)* it should achieve the ceiling of performance of s-d thresholding. If its effectiveness is far from optimal or unsatisfactory, further improvements of s-d thresholding should come from using alternative mixtures or training data. Nevertheless, some other mixtures are computationally more difficult—or even impossible—to compute.

References

[1] A. Arampatzis. Unbiased s-d threshold optimization, initial query degradation, decay, and incrementality, for adaptive document filtering. In *TREC*, 2001.

[2] A. Arampatzis and A. van Hameren. The score-distributional threshold optimization for adaptive binary classification tasks. In *Proceedings SIGIR'01*, pages 285–293, 2001.

[3] A. Arampatzis, J. Beney, C. H. A. Koster, and T. P. van der Weide. Incrementality, half-life, and threshold optimization for adaptive document filtering. In *TREC*, 2000.

[4] A. Arampatzis, J. Kamps, M. Koolen, and N. Nussbaum. Access to legal documents: Exact match, best match, and combinations. In *TREC*. NIST, 2007.

[5] C. Baumgarten. A probabilitstic solution to the selection and fusion problem in distributed information retrieval. In *Proceedings SIGIR '99*, pages 246–253. ACM Press, 1999.

[6] A. Bookstein. When the most "pertinent" document should not be retrieved – an analysis of the Swets model. *Information Processing and Management*, 13(6):377–383, 1977.

[7] K. Collins-Thompson, P. Ogilvie, Y. Zhang, and J. Callan. Information filtering, novelty detection, and named-page finding. In *TREC*, 2002.

[8] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[9] K. H. Knuth. Optimal data-based binning for histograms, 2006. URL `http://arxiv.org/abs/physics/0605197v1`.

[10] D. D. Lewis. Evaluating and optimizing autonomous text classification systems. In *Proceedings SIGIR'95*, pages 246–254. ACM Press, 1995.

[11] R. Manmatha, T. M. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In *Proceedings SIGIR'01*, pages 267–275, 2001.

[12] NIST/SEMATECH. e-handbook of statistical methods, 2008. `http://www.itl.nist.gov/div898/handbook/`.

[13] S. Robertson. On score distributions and relevance. In *Proceedings of 29th European Conference on IR Research, ECIR'07*, pages 40–51. Springer, Berlin, 2007.

[14] S. E. Robertson. The parametric description of retrieval tests. part 1: The basic parameters. *Journal of Documentation*, 25 (1):1–27, 1969.

[15] J. A. Swets. Information retrieval systems. *Science*, 141 (3577):245–250, 1963.

[16] J. A. Swets. Effectiveness of information retrieval methods. *American Documentation*, 20:72–89, 1969.

[17] Y. Zhang and J. Callan. Maximum likelihood estimation for filtering thresholds. In *Proceedings SIGIR'01*, pages 294–302. ACM Press, 2001.