Using Models of Score Distributions in Information Retrieval

R. Manmatha, F. Feng and T. Rath [*]
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts
Amherst, MA 01003
manmatha@cs.umass.edu

ABSTRACT

Empirical modeling of a number of different text search engines shows that the score distributions on a per query basis may be fitted approximately using an exponential distribution for the set of non-relevant documents and a normal distribution for the set of relevant documents. This model fits not only probabilistic search engines like INQUERY but also vector space search engines like SMART and also LSI search engines. The model also appears to be true of search engines operating on a number of different languages. This leads to the hypothesis that all 'good' text search engines operating on any language have similar characteristics. The question then arises as to whether the shape of the score distributions reflects some underlying model of language or the search process itself. We discuss how they arise given certain assumptions about word distributions in documents.

We then show that given a query for which relevance information is not available, a mixture model consisting of an exponential and a normal distribution can be fitted to the score distribution. These distributions can be used to map the scores of a search engine to probabilities.

This model has many possible applications. For example, the outputs of different search engines can be combined by averaging the probabilities (optimal if the search engines are independent) or by using the probabilities to select the best engine for each query. Results show that the technique performs as well as the best current combination techniques. A number of different IR tasks may benefit from score modeling including filtering, multi-lingual retrieval and relevance feedback. We also discuss possible future improvements to the process of score modeling.

1. INTRODUCTION

In the 1960's and 70's, Swets [10] and other researchers attempted to model the score distributions of search engines for relevant and non-relevant documents and then use statistical detection theory to find a threshold to decide what was relevant. This approach seems to not have been successful for a couple of reasons. First, the choice of the models seems to have been wrong; they assumed that the scores of both the relevant and non-relevant documents could be modeled using Gaussians. For example, van Rijsbergen [11] commented that for search engines like SMART there was no evidence that the two distributions were similarly distributed let alone normally. Second, computing these score distributions required knowledge of the relevance of at least some set of the documents. While this may be reasonable for a filtering task, in general, relevance information is not available for most information retrieval tasks.

Here we discuss how score distributions for a given query may be modeled using an exponential distribution for the set of non-relevant documents and a normal distribution for the set of relevant documents. We further show that when relevance information is not available, these distributions can be recovered by fitting a mixture model with a Gaussian and an exponential component to the output scores of search engines on a per query basis. This novel approach to score modeling is then used to map the scores to probabilities using Bayes' Rule. Note that no training is required for this approach and in addition no assumption is made on the kind of search engine used. The model has been shown to work for a large number of search engines on TREC-3 and TREC-4 data including probabilistic search engines like INQUERY and vector space search engines like SMART. This model has also been shown to work for other engines like the LSI search engine and the score distributions of TREC-6 INQUERY and SMART data on Chinese. To our knowledge, this is the first attempt at recovering the relevant and non-relevant distributions when no relevance information is available.

We hypothesize that all 'good' text search engines operating on any language have similar characteristics. This leads to the question of whether these score distributions reflect some underlying model of language itself. We discuss how the shape of the score distributions arise given certain assumptions about word distributions in documents.

The probabilities of relevance obtained from this model have many possible applications. For example thresholds for filtering may be selected using this approach or the probabilities may be used to combine the search from many distributed databases or multi-lingual or multi-modal databases. Here, we briefly discuss how we can combine the outputs of different search engines (the meta-search problem). for more details see [6].

The rest of the paper is divided as follows. The next section discusses related work. This is followed by Section 3 which discusses the modeling of score distributions of relevant and non-relevant

[*] This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the National Science Foundation under grant number IRI-9619117, in part by NSF Multimedia CDA-9502639 and in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor(s).

documents and how these distributions may be recovered in the absence of relevance information by using a mixture model. Solving for the mixture model using Expectation-Maximization (EM) is also discussed. Finally, Bayes' Rule is used to map the scores to probabilities of relevance. Section 4 discusses the theoretical intuition behind using such models. Section 5 discusses how the model and the probabilities derived from it can be used for evidence combination. Section 6 discusses possibilities of future work in this area. Finally, Section 6 concludes the paper.

2. RELATED WORK

In related and independent work Arampatzis et al [1] discussed how the relevant scores could be modeled using a Gaussian distribution and the non-relevant scores using an exponential distribution. They applied this to selecting the threshold dynamically for a filtering application where the relevance of a small subset of the data is known ahead of time. This subset can then be used to derive the relevant and non-relevant distributions separately. Zhang and Callan [12] argued that the distribution recovered by Arampatzis et al were biased and suggested ways to improve the estimation. We note that both of these papers require relevance information for some subset of the data.

For more discussions of related work on previous work on score modeling and on combining different search engines see [6]..

3. MODELING SCORE DISTRIBUTIONS OF SEARCH ENGINES

In this section we describe how the outputs of different search engines were modeled using data from the text retrieval conferences (TREC). TREC data provides the scores and relevance information for the top 1000 documents for different search engines. For the experiments here data from the ad hoc track of the TREC-3 and TREC-4 for a number of different search engines was used. We will show examples of the modeling on queries from INQUERY INQUERY is a probabilistic search engine from the University of Massachusetts, Amherst.

There are 50 queries available with document scores and relevance information for each query. We examine the relevant and non-relevant data separately. The data are first normalized so that the minimum and maximum score for a query are 0 and 1 respectively.

Figure 1 shows a histogram of scores for query 151 from TREC-3 for a set of non-relevant documents. The histogram clearly shows the rapid fall in the number of non-relevant documents with increasing score. A maximum-likelihood fit of an exponential curve to this data is also shown. For the purposes of fitting the exponential, the origin is shifted to the document with the lowest score. Figure 2 shows a histogram of scores for the set of relevant documents for the same query. The histogram approximates a normal distribution. The plot also shows a maximum-likelihood fit using a Gaussian with mean 0.466 and variance 0.042. The exponential fit previously obtained for the non-relevant documents is also plotted in the figure.

The same process was repeated for all 50 queries in this track and in most of those cases it was found possible to fit the non-relevant data with exponentials and the relevant data using Gaussians.

We have so far been able to fit parametric forms to the score distributions given relevance information. When running a new query, however, relevance information is not available. Clearly, it would be useful to fit the score distributions of such data. A natural way to do this is to fit a mixture model of a shifted exponential and a Gaussian to the combined score distribution. This approach is discussed in the next section.

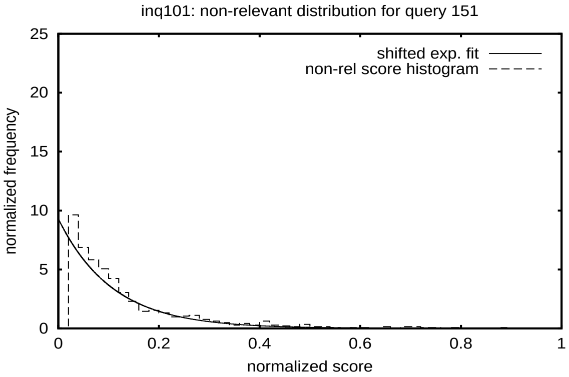

Figure 1: Histogram and shifted exponential fit to non-relevant data for query 151 INQUERY (inq101)

3.1 Mixture Model Fit

Consider the situation where a query is run using a search engine. The search engine returns scores but there is no relevance information available. We show below that in this situation, a mixture model consisting of an exponential and a Gaussian may be fitted to the score distributions. We can then identify the Gaussian with the distribution of the relevant information in the mixture and the exponential with the distribution of the non-relevant information in the mixture. Essentially this allows us to find the parameters of the relevant and non-relevant distributions without knowing relevance information apriori.

The density of a mixture model $p(x)$ can be written in terms of the densities of the individual components $p(x|j)$ as follows: [2].

$$p(x) = \sum_j P(j)p(x|j) \tag{1}$$

where j identifies the individual component, the $P(j)$ are known as mixing parameters and satisfy $\sum_{j=1}^{2} P(j) = 1, 0 \leq P(j) \leq 1$. In the present case, there are two components, an exponential density with mean λ

$$p(x|1) = \lambda \exp(-\lambda x) \tag{2}$$

and a Gaussian density with mean μ and variance σ^2

$$p(x|2) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{3}$$

A standard approach to finding the mixing parameters and the parameters of the component densities is to use Expectation Maximization (EM) [2]. This is an iterative procedure where the Expectation and Maximization steps are alternated. Space precludes us from discussing the details of the EM algorithm and the update equations used. The reader is referred to [2] for a good introduction to EM.

The procedure needs an initial estimate of the component densities and mixing parameters. Given that, it rapidly converges to a solution. Using EM to fit the data gives the mixture fit shown in Figure 3. The figure plots the mixture density as well as the component densities for the exponential and Gaussian fits. For comparison Figure 2 shows the exponential and Gaussian fits to the non-relevant and relevant data. Comparing the two figures, it appears that the strategy of interpreting the Gaussian component of the mixture with the relevant distribution and the exponential component of the mixture with the non-relevant distribution is a reasonable one. We should note that the correspondence between the mixture components and the fits to the relevant/non-relevant data is not always as good as that shown here but in general it is a reasonable fit.

This model has been fitted to a large number of search engines on TREC-3 and TREC-4 data including probabilistic engines like INQUERY and CITY and a vector space engine (SMART) as well as Bellcore's LSI engine. The fit appears to be better for "good" search engines (engines with a higher average precision in TREC-3) and worse for those with a lower average precision. The model has also been able fitted to document scores for searches on INQUERY and SMART indexing a Chinese database.

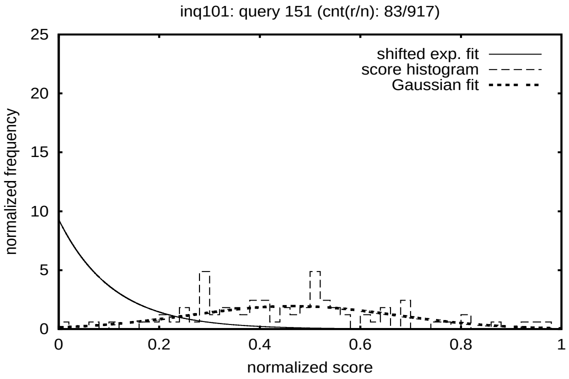

Figure 2: Exponential fit to non-relevant data and Gaussian fit to relevant data for query 151 INQUERY (inq101)

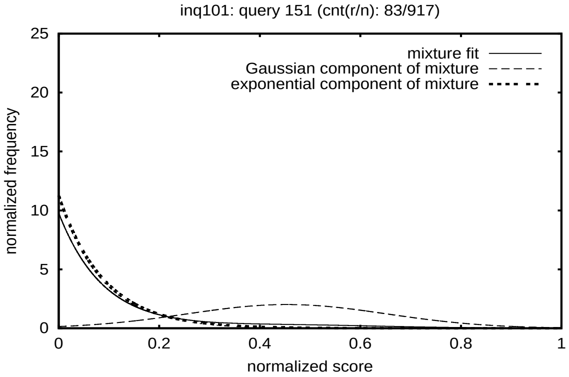

Figure 3: Mixture model fit showing exponential component, Gaussian component and the combined mixture for query 151 INQUERY (inq101). Compare with Figure2

3.2 Computing Posterior Probabilities

Using Bayes' Rule one can compute the probability of relevance given the score as

$$P(rel|score) = \frac{P(score|rel)P(rel)}{P(score|rel)P(rel) + P(score|nonrel)P(nonrel)} \tag{4}$$

where $P(rel|score)$ is the probability of relevance of the document given its score, $P(score|rel)$ and $P(score|nonrel)$ are the probabilities of score given that the document is relevant and score given that the document is non-relevant respectively. P(rel) and P(nonrel) are the prior probabilities of relevance and non-relevance.

In our model, $P(score|rel)$ is given by the Gaussian component of the mixture while $P(score|nonrel)$ is given by the exponential part of the mixture. P(rel) and P(nonrel) may be obtained by using the mixing parameters. Thus, $P(rel|score)$ can be computed in a simple manner. There are a number of considerations in computing the posterior including making sure that it is monotonic. For further details see [6]

3.3 Comments on Fitting Distributions and Mixture Models

There is a large family of densities which could possibly have fit the data. For example, the Poisson and Gamma distributions approximate the Gaussian for appropriate parameter choices. However, using a Poisson/Poisson (non-relevant/relevant) or an exponential/ Poisson combination did not fit the data well. On the other hand, while an exponential/Gamma fit the non-relevant and relevant data when separately fitted, a mixture fit with exponential and Gamma components did not converge to the right answer. In this case the Gamma component also converged to an exponential (the exponential density is a special case of the Gamma function). Thus our choice of distributions - exponential for the non-relevant and Gaussian for the relevant - is dictated by the consideration that the functions fit the data well and by the consideration that they can be recovered using a mixture model when relevance information is not available.

4. SHAPE OF DISTRIBUTIONS

We will now attempt to give some insight into the shape of the score distributions.

Given a term (or word) assume that the distribution of this term in the set of relevant documents is given by a Poisson distribution with parameter λ_r. That is,

$$P_r(x) = \frac{\lambda_r^x \exp(-\lambda_r)}{x!} \tag{5}$$

where x is the number of times that the term occurs in a particular document and $P_r(x)$ is the probability of x occurrences of the term in the set of relevant documents. Also assume that its distribution in the set of non-relevant documents is given by another Poisson distribution with the parameter λ_n and let $P_n(x)$ be the probability of x occurrences of the term in the set of non-relevant documents. In general, λ_n will be much smaller than λ_r.

Numerous attempts have been made to model word distributions in the past. Harter [4] used a mixture of 2 Poissons to model the distributions of words in a document. Our model in this section is closely related to his model. It has been argued by some researchers that the 2 Poisson model is not a good approximation and that other distributions like the negative binomial are better models of the distributions of words in documents [8]. Since we would like to fit a distribution to the relevant and another to the non-relevant, it is much more convenient for us to assume the 2-Poisson model here. Additionally, the main purpose of this section is to provide some insight and not a rigorous derivation.

Given a query consisting of 1 term and assuming that the score given to a document is proportional to the number of matching words in the document, the distribution of the scores of relevant documents is then given by the Poisson distribution:

$$P_r(x) = \frac{\lambda_r^x \exp(-\lambda_r)}{x!} \qquad (6)$$

and the distribution of the scores of non-relevant documents $P_n(x)$ is given by another Poisson distribution: The actual scores for many search engines is weighted by some function of the term frequency and the inverse document frequency. However, empirical evidence [3] shows that the score may be reasonably approximated as being proportional to the number of matching words.

For the set of relevant documents, λ_r will usually be large. For large values of λ, the Poisson distribution tends to a normal distribution (see Figure 4). On the other hand for small values of λ, the Poisson distribution will tend towards a distribution which is falling rapidly (see Figure 4). The shape of these curves is consistent with the experimental modeling of scores for TREC-3 and TREC-4 data (see the previous section). The experiments showed us that the normal distribution is a good fit for the score distributions of the relevant data. For non-relevant data, the experiments show that the exponential distribution is a good fit. For small λ_n, the Poisson distribution shows a decreasing distribution. Although, this is not the same as an exponential distribution, it does have the same general shape as an exponential (rapid monotonic decrease).

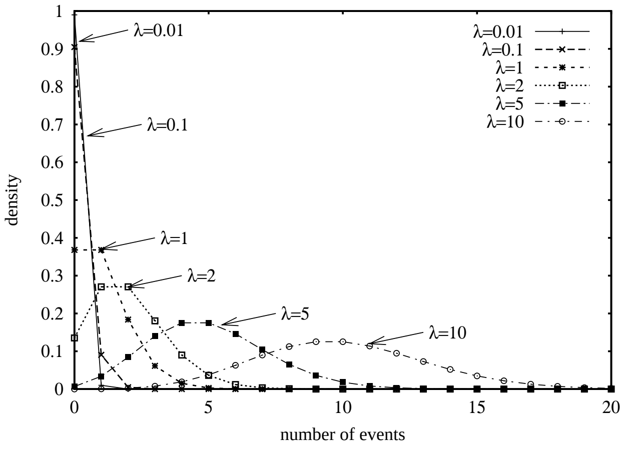

Figure 4: The Poisson distribution for different values of λ.

It is much harder to derive the score distributions if the query consists of two or more terms. This is because the actual scores of search engines are complicated functions. However, there is empirical evidence that the major contribution to the scores is provided by the number of matching terms [3]. We also note that Robertson and Walker [9] motivated a $tf - idf$ scoring function from the 2-Poisson model. We assume first that the score is proportional to the number of matching words and provide an intuitive argument for queries with two or more terms. For simplicity we will consider the case where the query has just two terms but the argument applies in general. In this case we can assume that the two terms say “oil” and “spill” can be clubbed together to create a single term - “oil spill”. Then the λ_r (the average frequency of a term over relevant documents) for joint occurrences of this term “oil spill” is much lower than the λ_r for either “oil” or “spill”. In other words the chances that the terms “oil spill” occur together is much less than that of finding “oil” or “spill”. When the query contains two terms, it is reasonable to assume that the λ_n (i.e. the average frequency over non-relevant documents) does not change much as it essentially reflects the background probabilities of the word.

The Poisson model for the shape of the relevant and non-relevant distributions that we have derived applies to both probabilistic engines like INQUERY and vector space engines like SMART. For vector space engines the number of matching terms is given by the dot product of two vectors - one representing the query and the other the document. Further, this model is language independent (as long as word frequencies in any language have an approximately 2-Poisson like distribution). Thus, we predict that a mixture of exponential and Gaussian distributions will fit a much larger class of text search engines operating on different languages.

5. COMBINING SEARCH ENGINES

The posterior probabilities obtained by using the model discussed above has many possible applications. For example the probabilities could be used to select a threshold for filtering documents or for combining the outputs in distributed retrieval. Here we discuss one possible application which involves combining the outputs of different search engines on a common database to improve results.

Some approaches to combining score distributions have focused on normalizing the range of the scores and then combining them by simple techniques like linear combination or by taking the minimum and maximum scores. However, a simple (linear) range normalization does not take into account the actual distribution of the scores.

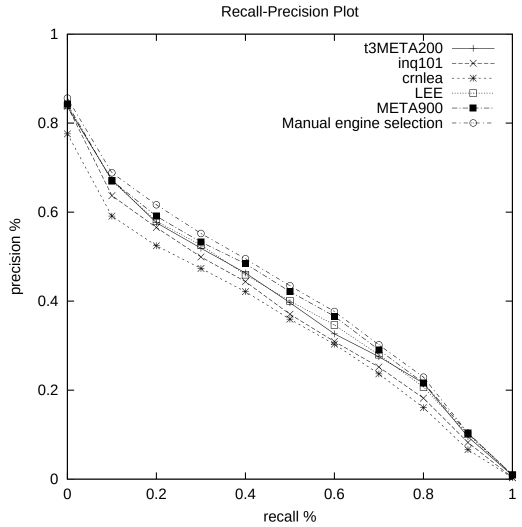

Figure 5: Recall precision graphs for combining inq101 and crnlea using different techniques (see text). Data from TREC-3

There are a number of possible ways the probabilities can be used to combine the search engines. We propose two alternative schemes for combination. The first involves averaging the probabilities. This is optimal in the sense of minimizing the Bayes’ error if the search engines are independent. Of course the outputs of search engines are not necessarily independent. In the following discussions, data are taken from TREC-3, INQUERY and SMART are the individual engines to be combined, META200 denotes combination by averaging the posteriors obtained using the mixture model, while META900 denotes the combination by averaging the posterior probabilities using the Gaussian and exponential fits assuming relevance is given. Thus, any difference between META200 and META900 is caused by the errors in performing a mixture fit to the model. LEE denotes Lee's COMBMNZ [5] technique which is one of the best ad hoc (in the sense of being empirically motivated) techniques around. LEE's technique involves normalizing the socre of each engine on a per query basis. The score of each document in the combination is then obtained by multiplying this sum by the number of engines which had non-zero scores (COMBMNZ). COMBMNZ is equivalent to weighted averaging. The manual engine selection technique involves selecting the best engine(s) and discarding the worst engine(s) on a per query basis using the average precision for that query. Manual engine selection provides an indication of the best combination result we can achieve. Note that both META900 and manual engine selection require relevance information and are only plotted to provide a baseline for understanding the limits of combination.

Figure 5 shows recall-precision plots for combining INQUERY and SMART on TREC-3 data. Precision is defined as the fraction of retrieved documents which are relevant while recall is the fraction of relevant documents which have been retrieved. The recall-precision graph is usually created by averaging over a certain number of queries - in this case 50. As the figure shows META200 performs considerably better than either INQUERY and SMART - in fact about 6% better than INQUERY and 13% better than SMART. LEE is slightly better (about 1%) than META200 although the difference is not significant. META900 has an average precision about 10% better than INQUERY and clearly performs better than either META200 or LEE's implying that if the mixture fit could be improved the technique would perform even better. Finally, the plot for manual engine selection clearly indicates that both META200 and LEE's are close to obtaining the best performance possible from combination.

Figure 6 describes combination results for the top five engines in TREC-3. The x-axis is the number of engines combined while the y-axis is the average precision. As the plot clearly shows combination clearly improves the results. There are four graphs in the figure. The first curve is the average precision of the individual search engines. The second plot META200 shows the combination method applied to 1, 2, 3, 4 or 5 engines. As can be clearly seen there is a considerable improvement over using even the best search engine and overall the improvement seems to increase with the number of search engines combined. With the top 2 engines, META200 shows an improvement of 6% over the best single engine and using the top 3 engines, META200 shows an improvement of almost 12%. LEE's COMBMNZ technique is also shown in the same graph. It's average precision is seen to be slightly worse than META200 but the difference is not really significant. The performance obtained using META900 (i.e. combination with the posterior probabilities obtained with relevance information) is 15% better than the best single engine. Again this indicates that if the mixture fit were improved we could do even better.

This approach to combination works for other engines and on documents operating on other languages. For more details see [6].

5.1 Automatic Engine Selection

Search engines do not perform consistently for every query. Thus, for some queries discarding the results of a particular engine might actually improve the combination. The manual engine selection strategy actually does this. If we could figure out how to select the engines automatically, we might be able to achieve better performance than using just the average. In particular when the performance of one search engine is much worse than the other, averaging is not necessarily the best strategy (see [6]). The ability to model and compute the relevant and non-relevant distributions allows us to develop techniques to automatically select engines on a per query basis. A number of different approaches can be tried. Space precludes us from discussing the details (see [6]. We give below the average precision numbers for different approaches to combining the results from INQUERY and SMART on TREC-3. search engines. Note that automatic engine selection is the best automatic technique, although the differences between it, Lee's and META200 are probably not significant. INQUERY - 0.3659, SMART - 0.3419, META200 - 0.3876 LEE - 0.3915, Automatic Engine selection - 0.3968, META900 0.4047 Manual Engine Selection - 0.4144

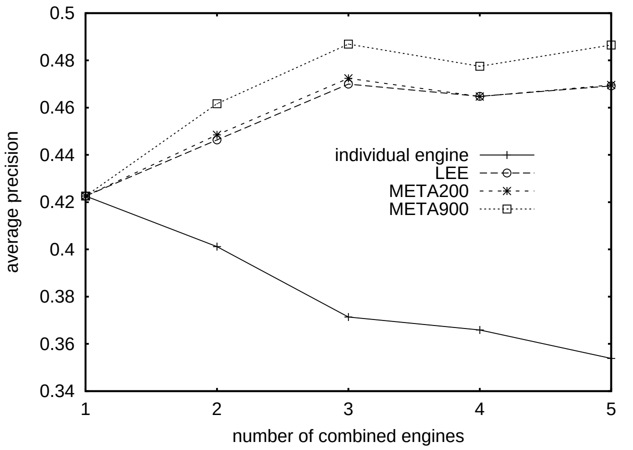

Figure 6: Recall precision graphs for combining the best five techniques from TREC-3.

5.2 Discussion of Combination Results

The results above show that the mixture modeling performs as well as the best current techniques (LEE) available for combination. There is scope for a slight improvement in estimating the mixture parameters as well using that for obtaining better combination. Of course it is also clear that we are approaching the limits of the best performance we can achieve.

The model for combination proposed here is more intuitively satisfying for a number of reasons. First, it combines engines in a natural way using probabilities and is therefore easier to explain. Second, it indicates where improvements can be made for better performance. Third, the same technique may be used for combining multi-lingual engines. It will also extend to multi-modal engines provided the distributions of scores behave in a similar way for search engines indexing other media.

6. WHITHER SCORE MODELING

The model for score distributions leads to a number of interesting questions and possible future applications and extensions of the technique.

The first interesting point to note is the large number of different search engines which seem to have similar score distributions. It is not obvious that this should be the case. For example, there are a large class of functions which could change the score distribution without changing the actual document rankings. This leads us to believe that the form of the score distributions is intimately related to the distributions of terms (words) in languages. We have intuitively justified the form of the score distributions from a two Poisson model for word distributions. However, a more formal derivation from either a Poisson model or a more accurate model of word distributions (say a negative binomial distribution) would be useful.

The assumption that the relevant and non-relevant distributions are given by a Gaussian and an exponential are only approximations. There are probably better models around and it would be interesting to find these. The use of any such models must take into account whether they can be recovered from a mixture when relevance information is not available. Thus, for example we have noticed that the gamma distribution did fit the separate relevant and non-relevant distributions better but the mixture model approach did not work when a mixture of two gamma distributions was used.

We note that the distributions obtained from the EM based mixture models are not identical to those obtained when complete relevance information is available. Further, the results of combining search engines using the EM based mixture model are also not as accurate as those obtained when complete relevance information is available. This suggests that there is room for improvement in the EM based mixture modeling. It also suggests the possibility of using partial relevance information (as in relevance feedback or even filtering) to improve the mixture modeling. In fact, relevance feedback may fit naturally into this model.

In the case of combination, one surprising result is that Lee's technique (based on normalizing and weighted averaging of the scores) performs as well as it does. Lee's technique was proposed on heuristic grounds. But its performance leads one to suspect that it actually approximates a more formal combination technique and it would be interesting to see if we can derive a more principled approach to Lee's technique.

The score modeling work can be applied to a number of different areas besides the meta-searching of a single database proposed in our SIGIR paper. An obvious extension is to the problem of combining results from multiple search engines which may operate on different databases. It is not completely obvious that the results will improve in this case. For example, one of the explanations provided for the efficacy of Lee's technique is that it tends to promote those documents which are returned by multiple search engines. When the databases are different, the same document is unlikely to be present in multiple databases and hence the efficacy of the proposed combination technique in such cases is unknown.

Other extensions could include the application to one or more databases in different languages. Multi-lingual retrieval often involves using the same query in multiple languages to search databases in different languages. A common problem in such situations is how the ranks output for search engines operating on different languages should be combined and there does not seem to be solution which is satisfactory. Score modeling could allow us to convert the scores of each search engine to a probability and then combine the probabilities. The probabilities can be combined in a more natural way We hope to do some experiments in the near future to test this hypothesis.

Recently [1, 12] score modeling has been applied to select filtering thresholds. However, this work has been done without using the mixture modeling approach - i.e. the Gaussian and exponential distributions are obtained using relevance information on a subset of the data. It may be worth examining whether the mixture modeling would improve the results.

Another interesting question that comes to mind is whether these models also fit search engines operating on other media. This needs to be tested. If true, it could lead to one way of combining multimedia engines for multi-modal retrieval.

7. CONCLUSION

We have demonstrated how to model the score distributions of a number of text search engines. Specifically, it was shown empirically that the score distributions on a per query basis may be fitted using an exponential distribution for the set of non-relevant documents and a normal distribution for the set of relevant documents.

It was then shown that given a query for which relevance information is not available, a mixture model consisting of an exponential and a normal distribution may be fitted to the score distribution. These distributions were used to map the scores of a search engine to probabilities.

The model of score distributions was used to combine the results from different search engines to produce a meta-search engine. The results were substantially better than either search engine provided no "search engine" performed really poorly. Different combination techniques were proposed including averaging the posterior probabilities of the different engines as well as using the probabilities and distributions to selectively discard some engines on a per query basis.

Future work will include attempts to further improve the modeling for better performance. Other possible applications of modeling score distributions like filtering will also be examined. Finally we will also examine the possibility that search engines indexing other media like images can also be modeled in the same way.

8. ACKNOWLEDGEMENTS

The origins of this work can be traced back to a discussion at the University of Glasgow. The first author would like to thank Bruce Croft for extensive discussions and also for pointing out relevant literature especially previous literature on score modeling in information retrieval.

9. REFERENCES

[1] C. H. A. K. A. Arampatzis, J. Beney and T. P. van der Weide. Incrementality, half-life and threshold optimization for adaptive document filtering. *To appear in the Proc. of the 9th Text Retrieval Conference (TREC-9)*.

[2] W. Greiff. The use of exploratory data analysis in information retrieval research. In W. B. Croft, editor, *Advances in Information Retrieval*, pages 37–72. Kluwer Academic Publishers, 2000.

[3] S. P. Harter. A probabilistic approach to automatic keyword indexing. *Journal of the American Society for Information Science*, 20:197–206, 1975.

[4] J. H. Lee. Analyses of multiple evidence combination. In *the Proc. of the 20th ACM SIGIR conf. (SIGIR'97)*, pages 267–276, 1997.

[5] R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. *To appear in the Proc. of the 24th ACM SIGIR conf.*, Sept 2001.

[6] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley, 2000.

[7] F. Mosteller and D. Wallace. *Inference and Disputed Authorship: The Federalist*. Addison Weseley, 1964.

[8] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *the Proc. of the 17th ACM SIGIR conf.*, pages 232–241, 1994.

[9] J. A. Swets. Information retrieval systems. *Science*, 141:245–250, 1963.

[10] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[11] Y. Zhang and J. Callan. Maximum likelihood estimation for filtering thresholds. *To appear in the Proc. of the 24th ACM SIGIR conf.*, Sept 2001.