Standard Deviation as a Query Hardness Estimator*

Joaquín Pérez-Iglesias and Lourdes Araujo

Universidad Nacional de Educación a Distancia
Madrid 28040, Spain
joaquin.perez@lsi.uned.es, lurdes@lsi.uned.es

Abstract. In this paper a new *Query Performance Prediction* method is introduced. This method is based on the hypothesis that different score distributions appear for 'hard' and 'easy' queries. Following we propose a set of measures which try to capture the differences between both types of distributions, focusing on the dispersion degree among the scores. We have applied some variants of the classic standard deviation and have studied methods to find out the most suitable size of the ranking list for these measures. Finally, we present the results obtained performing the experiments on two different data-sets.

1 Introduction

Query Performance Prediction (QPP) deals with the problem of estimating the difficulty of a query, where the difficulty degree is commonly measured in terms of the average precision (AP) obtained by the query. Thus, a query which obtains a low AP value is considered as a 'hard' query, while one with a high AP value would be considered as an 'easy' query. A classic application of QPP which has shown some interesting results is *selective query expansion* [1], where the quality prediction is applied to avoid the automatic expansion of those queries which would worsen the overall retrieval quality.

This paper introduces a novel approach for Query Performance Prediction, which falls into the so-called post-retrieval prediction methods. This type of predictors makes use of the information supplied by the search system, once the search has been performed, while pre-retrieval predictors compute the estimation before completing the search. Our approach is focused on the study of the scores assigned to the documents returned by a search system in response to a query. It is based on the hypothesis that there exist differences between the scores distribution of 'hard' and 'easy' queries. The dispersion in the scores of the document ranking list is measured in order to predict the query performance.

* This paper has been funded in part by the Spanish MICINN projects NoHNES (Spanish Ministerio de Educación y Ciencia - TIN2007-68083) and by MAVIR, a research network co-funded by the Regional Government of Madrid under program MA2VICMR (S2009/TIC-1542). Authors want to thank Álvaro Rodrigo-Yuste for his review and comments.

E. Chavez and S. Lonardi (Eds.): SPIRE 2010, LNCS 6393, pp. 207–212, 2010.

The rest of this paper is organised as follows. In Section 2 related work in query performance prediction is introduced, with a special emphasis on post-retrieval approaches. Then, in Section 3, a detailed description of the starting hypothesis and the different measures employed to compute the ranking list dispersion is given. Section 4 deals with the specific evaluation performed and the analysis of the obtained results. Finally, the main conclusions are given in Section 5.

2 Related Work

In the last years several techniques dealing with Query Performance Prediction have been proposed. The different prediction methods are usually classified into two main categories: *a*) *pre-retrieval* approaches, which try to estimate query difficulty without using the list of documents obtained from the search engine; *b*) and *post-retrieval*, which use the information obtained after submitting the query to the search engine.

Focusing on post-retrieval methods we can found the classic Clarity Score by Cronen-Townsend et al.[2]. Clarity Score tries to measure the ambiguity of a query with respect to the document collection. The ambiguity of a query is calculated using the Kullback-Leibler divergence (KLD) between the language models of the collection and the top ranked documents. A well performing query would show a high divergence value. Some methods based on Clarity Score appear subsequently, such as *Ranked List Clarity Score*, which replaces ranking scores by the document ranking position, and *Weighted Clarity Score*, which allows to assign a different weight to each query term in order to calculate KLD, both in [3].

Recently a new improved version of Clarity Score has been presented by Hauff et. al [4], which outperforms the original Clarity Score on performance prediction accuracy. A related approach to the one introduced here, where the scores of the ranking list are analysed, was developed by Diaz [5], who applied the similarity between the scores of topically close documents as a prediction value. A similar approach was proposed by Vinay [6], where the prediction is based on the correlation between the actual rank of a document and a computed expected rank, where the expected rank is obtained by modelling the score of a document as a Gaussian random variable.

The following section introduces the proposed post-retrieval predictor based on the scores dispersion in a ranking list.

3 Ranking List Score Dispersion as a Predictor

We based our approach on the hypothesis that some differences between document score distribution for 'hard' and 'easy' queries should be observed. For example, if a ranking list has a high value of dispersion in their document scores, it could indicate that the ranking function has been able to discriminate between relevant and non-relevant documents. On the other hand, if a low level of dispersion appears, it can be interpreted as if it was not able to distinguish between relevant documents from non-relevant ones.

This behaviour can be observed assuming a 'perfect probabilistic model', where documents are scored with a probability of relevance equal to either 1 or 0. Relevant documents are weighted with 1, while 0 is assigned to those considered non-relevant. In this theoretical model, the dispersion among scores will be maximised when an equal number of relevant and non-relevant documents are included within the ranking list. An immediate consequence of the application of dispersion as a query hardness estimator, is the importance of selecting a suitable size k for the document ranking list. The dispersion measured at a wrong ranking list size, which for example includes too many non-relevant documents or not enough relevant documents, will imply a misleading estimation of the query hardness.

An example of the differences in terms of score dispersion can be observed in Figure 1, where the five best (left) and five worst (right) performing queries from Robust 2004 track are represented. As it can be seen in the figure, best queries show a longer distance between maximum and minimum score and a sharp slope. However, queries with a poor performance show a higher similarity in their scores along the ranking list, a softer slope and a smaller distance between maximum and minimum scores.

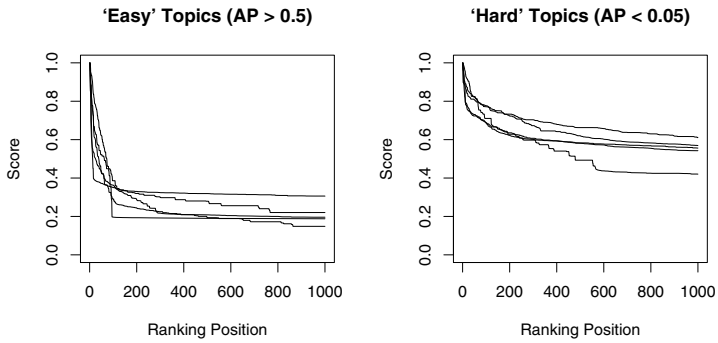

Fig. 1. 5 Best Performing Queries (left) vs 5 Worst Performing Queries (right), from Robust 2004 using BM25 as ranking model. Scores have been normalised in $[0, 1]$, dividing each score by the highest one. The maximum number of retrieved documents is fixed to 1000.

Concerning score distributions, previous works have tried to define how document scores are distributed along a ranking list. In general, it can be assumed that an appropriate model could be a mix of an exponential and a gaussian probability distribution. Exponential for non-relevant documents, and gaussian for relevant documents [7]. Usually most of the retrieved documents are non-relevant (exponential distribution), thus it is likely that a big majority of documents will obtain a low score.

Therefore, the shape of the ranking list exhibits a 'long tail' where most of the non-relevant documents are placed, see Figure 1. An overall effect of the ranking list 'long tail' is a reduction of the accuracy of the applied measures.

Next section describes the measures defined to capture the differences between 'hard' and 'easy' queries.

Proposed Measures

We have tested different measures for capturing the dispersion from the scores of a ranking list. All the proposed measures are based on standard deviation, but with some differences in the techniques used for fixing the most suitable ranking list size k. It should be noted that document scores are normalised by the maximum to allow a fair comparison between queries with different lengths.

Maximum Standard Deviation: This measure tries to minimize the effect of the *ranking list tail* by computing the standard deviation at each point in the ranking list and selecting the maximum value of the standard deviation thus found. Hence, those scores which appear at the *ranking list tail* have no influence in the calculation of the maximum standard deviation.

Standard Deviation at a Common Best k: Computing the standard deviation manually fixing its size k. With the selection of a suitable size k the noise introduced by the set of low scores is removed. Fixing k globally requires the selection of a common k value which maximises the correlation degree for all queries.

Estimating a Cut-Point k Automatically for each Query: The previously introduced measures establish a ranking list size k which is shared by all queries. Here we propose a method aimed at fixing the size k specifically for each query. For this estimation we use the number of documents which are retrieved when each term from the query is considered mandatory, which is equivalent to applying the boolean AND operator for all terms in the query. This estimator has been applied previously [4] to select an appropriate document set size in order to create a top ranking language model.

With the use of the AND operator some queries retrieve no documents at all, while for others the number of retrieved documents is large and similar to that obtained with the OR operator. In order to circumvent this situation a scaling linear function is applied. This function scales the original k value obtained from an AND search to a value closer to the expected median ($\tilde{k}$) for all queries, avoiding the undesired cases described above.

The linear transformation makes use of a free parameter λ, which defines how similar will be the new ranking list size to the original median. In order to ascertain the similarity between the original k value, obtained with the AND operator, and the scaled one k', obtained after the linear transformation, the next conditions have to be fixed:

$$\tilde{k'} = \tilde{k} \text{ and } k'_{max} = \lambda\tilde{k}$$

where k'_{max} is the longest ranking list size expected for the scaled values. Thus the linear transformation is calculated as $k' = ak + b$, where $a = \frac{(\lambda-1)\tilde{k}}{k_{max}-\tilde{k}}$, $b = (1-a)\tilde{k}$ and $\lambda \in [1, \frac{k_{max}}{\tilde{k}}]$.

4 Results

As usual the performance of the predictor is computed by measuring the correlation degree between the predictor and the real search system performance in terms of AP. A significant correlation degree between both measures means an accurate estimation of the query performance. For this purpose the Pearson and Kendall correlation coefficients have been applied. Both correlation coefficients calculate a real number in the range $[-1, 1]$, where 1 means perfect correlation, -1 means a perfect inverse correlation and 0 means no correlation at all. The different measures proposed in this paper have been tested with the set of documents from TREC Disk4 & 5 and GOV2 collections, including topics from 301 to 450 and 700 to 800. Only the field title from each topic has been employed for both collections and the Okapi BM25 [8] has been applied as ranking function.

The results obtained with both datasets are shown in Table 1. Each row corresponds to one of the proposed measures: (σ_{full}) the standard deviation for the whole ranking list (1000 for TREC 4 & 5 and 10000 for GOV2); (σ_{best}) the standard deviation at a common best size k for all topics (100 for TREC 4 & 5 and 1000 for GOV2); (σ_{max}) the maximum standard deviation; and (σ_k) the standard deviation at a specific size k for each query, using the automatic method proposed previously. The last two rows include the results obtained with *Clarity Score* (CS) and the *Improved Clarity Score* (ICS)[1] version proposed by Hauff et al. in [4].

In relation with the capability to capture dispersion of the different proposed measures. Here we can observe how the standard deviation of the whole ranking list obtains the lowest correlation value, as it was expected because of the noise introduced by the *ranking list tail*. The results obtained with the maximum standard deviation are similar to those achieved with the selection of a common optimal ranking list size, but with the advantage of removing the use of a parameter to cut the ranking list and thus avoiding the problem of computing this optimal common k. The best results have been achieved by fixing automatically a suitable ranking list size for each topic.

Table 1. Pearson and Kendall coefficients obtained with the proposed measures

	Pearson					**Kendall**				
	TREC 4 & 5			**GOV2**		**TREC 4 & 5**			**GOV2**	
	301-350	**351-400**	**401-450**	**701-750**	**751-800**	**301-350**	**351-400**	**401-450**	**701-750**	**751-800**
σ_{full}	0.3886	0.3358	0.4367	0.4551	0.1497	0.2721	0.2295	0.2261	0.2959	0.1530
σ_{best}	0.7455	0.5188	0.6363	0.3808	0.2509	0.4693	0.3690	0.3146	0.1921	0.2534
σ_{max}	0.6488	0.4298	0.7854	0.3522	0.2322	0.4761	0.3282	0.4659	0.1802	0.2380
σ_k [4]	0.7802	0.5623	0.7136	0.4957	0.3019	0.5340	0.3996	0.3639	0.3656	0.2193
CS	0.5390	0.3095	0.5727	0.6033	0.4441	0.4198	0.2172	0.3045	0.4149	0.3299
ICS	0.6330	0.5106	0.7064	0.5422	0.5498	0.4998	0.4002	0.5624	0.3723	0.4181

[1] CS and ICS results have been taken from [4].

Finally, the correlation found in the GOV2 collection is not so strong, as for TREC 4 & 5 dataset, although it is almost equivalent to the achieved with CS or ICS. We believe that this decrease is caused by the noise which appears in a Web environment, where for each query the number of retrieved documents that obtain a similar score is higher than for TREC 4 & 5. A similar problem was addressed by Hauff et al. [4] in the application of predictors to a Web environment.

5 Conclusions

A novel query performance predictor has been introduced in this paper. A high correlation value has been found using the proposed measures based on standard deviation, suggesting the hypothesis described on this work about the relation between the scores dispersion and query performance.

The application of the standard deviation as a dispersion measure for a ranking list has shown to be an effective approach if a suitable ranking list size is selected. We have introduced three different techniques for fixing this size, which have improved the results reducing the noise introduced by the *ranking list tail*. Best results have been achieved using a specific ranking list size per topic.

References

1. Amati, G., Carpineto, C., Romano, G., Bordoni, F.U.: Query Difficulty, Robustness and Selective Application of Query Expansion. In: McDonald, S., Tait, J.I. (eds.) ECIR 2004. LNCS, vol. 2997, pp. 127–137. Springer, Heidelberg (2004)
2. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR 2002. ACM Press, New York (2002)
3. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Precision prediction based on ranked list coherence. Information Retrieval 9(6), 723–755 (2006)
4. Hauff, C., Murdock, V., Baeza-Yates, R.: Improved query difficulty prediction for the web. In: CIKM 2008: Proceedings of the 17th ACM conference on Information and knowledge management, pp. 439–448. ACM, New York (2008)
5. Diaz, F.: Performance prediction using spatial autocorrelation. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR 2007, p. 583. ACM Press, New York (2007)
6. Vinay, V., Milic-Frayling, N., Cox, I.: Estimating retrieval effectiveness using rank distributions. In: CIKM 2008: Proceedings of the 17th ACM conference on Information and knowledge management, pp. 1425–1426. ACM, New York (2008)
7. Robertson, S.: On Score Distributions and Relevance. Advances in Information Retrieval 4425, 40–51 (2007)
8. Jones, K.S., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments. Inf. Process. Manage. 36(6), 779–808 (2000)

† Where λ was fixed to 5 for TREC 4 & 5 and 8 for GOV2.