DOI: 10.12700/AIS.2025.014

A Study on Predicting Term Weighting Models with Query Performance Predictors to Enhance Information Retrieval Effectiveness

1st Gökhan Göksel
Department of Computer Engineering
Eskis¸ehir Technical University
Eskis¸ehir, Turkey
gciplak@eskisehir.edu.tr

Abstract—Information retrieval systems often exhibit variability in effectiveness across queries and retrieval models. This study investigates the use of pre-retrieval query performance predictors to guide per-query prediction of term weighting models. A prediction-based method (PBM) is employed using a K-Nearest Neighbors (KNN) to predict between BM25 and DFRee term weighting models based on features derived from query performance predictors. Experiments are conducted on standard TREC datasets, which are CW09B, CW12B, and GOV2, and retrieval effectiveness is evaluated using nDCG@20. Results demonstrate that PBM consistently outperforms the baseline models. It achieves statistically significant improvements in average effectiveness and enhanced robustness across queries. Per-query analyses show that PBM accurately predicts the more effective model between term weighting models. These findings indicate that query-level model prediction based on query performance predictors can substantially improve IR performance.

Index Terms—index term weighting, query performance prediction, information retrieval

I. Introduction

Information retrieval (IR) systems are designed to address users' information needs. Ideally, the response of an IR system should fully satisfy those needs. Systems that consistently succeed in meeting users' diverse information requirements are regarded as robust. Such systems distribute their effectiveness evenly across all queries, rather than a marginal performance increment on only a subset [1].

An IR system typically consists of three sub-components for addressing users' information needs: preprocessing, term–document matching, and re-ranking. Among these, the term–document matching phase serves as the core of the system. Although the other components are relatively optional, they significantly influence the overall performance of an IR system [2], [3], at least as much as the core component itself [4]. From this perspective, the configurations chosen for each component can substantially affect the retrieval of documents for a given set of queries. Consequently, in practical appli- cations, IR systems often fail to satisfy all user information needs at a desirable level [5].

The literature has a long history of improving the effectiveness of IR systems across all sub-components. For instance, employing stemming in the pre-processing phase is a typical and substantial application, as stemming aims at alleviating the mismatching problem between query and document terms. Accordingly, both, rule-based [6]–[8] and statistical stemming methods [9], [10] have been widely integrated into IR systems [2], [11].

Term weighting models, as a core sub-component of the IR systems, have a central role in identifying relevant documents within a corpus. Because of its importance, a wide range of models have been developed and presented in the literature. BM25 [12], DFRee [13], and DFIC [14] are some example methods from the extensive literature. Each method relies on a distinct probability distribution for modeling the query term frequencies. BM25 employs a Two-Poisson distribution while DFRee uses a Hypergeometric distribution. On the other hand, DFIC utilizes the Empirical distribution. Besides these term weighting models, many other models rely on different kinds of distributions and techniques.

Having relevant documents ranked higher makes it easier for users to access the information they need more quickly. The third phase re-ranks the documents retrieved by the index term weighting model to promote documents considered more relevant to higher positions in the ranked list. Metrics such as Normalized Discounted Cumulative Gain (nDCG) [15], [16] explicitly account for ranking positions when evaluating retrieval effectiveness. Similar to previous phases, multiple methods have been explored for re-ranking. Learning-to-rank method employs machine learning methods to re-rank retrieved documents for a given query [17]. On this account, selected or used features based on query-independent or query-dependent gain importance for this phase. The proposed methods for re-ranking in the literature [17]–[19] inherently affect the measured retrieval performance.

The capabilities of large language models (LLMs) and pretrained language models (PLMs) in capturing semantic meaning have also influenced all sub-components of information retrieval. On this account, several retrievers such as DeepCT [20], DeepImpact [21], uniCOIL [22], and SPLADE [23] are introduced to retrieve documents for a given query from the corpus, while re-rankers such as MonoT5 and DuoT5

[24], and RankT5 [25] have been proposed for post-retrieval ranking in the literature. However, the computational cost of LLMs and PLMs has restricted their deployment to relatively small corpora compared to web-scale datasets in IR experiments. Consequently, an IR system includes multiple sub-components, and each sub-component affects the retrieval performance for each query differently. While some queries benefit from a particular system configuration, the same setup may hinder effectiveness for other queries.

In this study, an experiment is conducted to predict the appropriate term weighting model using several pre-retrieval query performance predictors. The prediction relies on the K-Nearest Neighbors (KNN) and aims at determining which of the two term weighting models yields more effective results. By employing two system setups (one for each term weighting model), the usefulness of query performance predictors in guiding model selection is examined. Furthermore, the retrieval effectiveness of this prediction-based method (PBM) is analyzed on a per-query basis.

II. Experiments

The experiments are conducted on Apache Lucene, an open-source search engine platform. Although its usage in the industry is well established, its adoption in academic research has also become increasingly widespread. For the experiments, the Web documents are stripped of the HTML tags to obtain textual content and then indexed in Lucene for search and retrieval. Standard tokenizer and lower case filtering are applied to the textual content without stemming during the indexing process.

A. Datasets

The ClueWeb09 collection consists of approximately 1 billion web documents gathered in January and February 2009. The collection's Category B subset contains approximately 50 million English web pages. The subset was used in TREC Web Tracks from 2009 to 2012 (CW09B). The ClueWeb12 comprises approximately 733 million English web documents, gathered between 10 February and 10 May 2012. A uniformly extracted sample of 7% of the collection is named Category B13. The subset collection was used in the TREC Web Tasks from 2013 to 2016 (CW12B). The GOV2 collection [26] involves approximately 25 million web documents. These documents were gathered from the .gov domain. The TREC Terabyte Tracks 2004–2006 (GOV2) study was conducted using this collection. The experiments are carried out on standard TREC datasets with their associated tracks. The numbers of queries in these tracks are listed in Table I

TABLE I
This lists the collections, the number of queries in the tracks, and the abbreviations used in the experiments.

Collection	Track	Abbr. in Study	# queries
CW09B	Web 2009, 2010, 2011 & 2012	CW09B	197
CW12B	Web 2013 & 2014 Tasks 2015 & 2016	CW12B	185
GOV2	Terabyte 2004, 2005 & 2006	GOV2	149

B. Query Performance Predictors

The variability in retrieval effectiveness across both queries and systems has motivated research into query performance prediction [27]. The objective is to estimate the quality of search results produced by a specified retrieval system for a given query without relying on human relevance judgments. Accurate prediction would enable information retrieval systems to better manage difficult queries and mitigate performance fluctuation. Furthermore, identifying the most effective retrieval strategy on a per-query basis has the potential to substantially improve overall effectiveness.

The potential of query performance prediction for improving IR effectiveness has motivated extensive research into diverse prediction methods. These approaches are typically classified into pre-retrieval and post-retrieval methods. Pre-retrieval methods estimate search result quality before retrieval. It relies solely on the raw query and term statistics available at indexing time. In contrast, post-retrieval methods leverage not only the query and term statistics but also features derived from the retrieved results. In this study, experiments are performed on only pre-retrieval predictors since using post-retrieval predictors requires multiple runs of queries on the IR systems. In this context, the employed predictors for the KNN are listed:

- Inverse document frequency (idf): Queries are usually consist of multiple terms. Mean, variance, and maximum *idf* values of query terms are produced for the feature set. The used formula: $idf = ln(N/df\ (t)$, where N and $df\ (t)$ refer to the number of documents in the collection and the document frequency of the term t
- Query Scope [28]: Represents the percentage of documents in the corpus that include at least one query term. The equation: $\omega = -log(n_Q/N\ )$, where n_Q refers to the number of documents that include at least one query term.
- Gamma [28]: Captures the distribution of information content across the component terms of a query.
  The equation: $\gamma = idf_{max}/idf_{min}$
- Pointwise Mutual Information (PMI) [29]: PMI scores for each query term pairs. Mean and maximum *PMI* values of query term pairs are produced for the feature set.
  The equation:

$$PMI(t_1, t_2) = \log \frac{Pr(t_1, t_2 \mid D)}{Pr(t_1 \mid D) \cdot Pr(t_2 \mid D)}$$

- Simplified Clarity Score (SCS) [30]: Quantifies query specificity by measuring the Kullback–Leibler divergence between the query language model and the collection language model.

TABLE II
AVERAGE RETRIEVAL RESULTS OF IR SYSTEMS IN THE METRIC NDCG@20.

	CW09B	CW12B	GOV2
BM25	0.1606	0.0781	0.3486
DFRee	0.1596	0.0891	0.3552
PBM	0.1651^{D}	0.0926^{B}	0.3601^{B}

The equation:

$$SCS(q) = \sum_{t \in q} Pr(t \mid q) \log \frac{Pr(t \mid q)}{Pr(t \mid D)}$$

- Contribution of a term to the total inertia (CTI) [31]: Total inertia reflects the overall magnitude of squared deviations from independence. Mean, variance, and maximum *CTI* values of query terms are produced for the feature set.
  The equation:

$$CTI = \frac{1}{N} \sum_{j=1}^{c} \frac{(tf_{ij} - e_{ij})^2}{e_{ij}}$$

  where *i* refers to term number and *j* refers to document number.
- Skewness and Kurtosis: Skewness and kurtosis of the frequency distribution for a query term. Mean and variance *Skewness* and *Kurtosis* values of query terms are produced for the feature set separately.
- Collection Query Similarity (SCQ) [32]: Measure the query similarity to the collection. Mean, variance, and maximum *SCQ* values of query terms are produced for the feature set.
  The equation: $SCQ(t) = (1 + \log(tf(t, D))) \cdot idf(t)$

C. Experimental Setup

The prediction decides which term weighting model should be applied to the information retrieval system for a given query. For this purpose, DFRee [13] and BM25 [12] are selected as baseline term weighting models. These are well-known and widely used models in literature and practical applications. Retrieval effectiveness is evaluated using nDCG at the top 20 documents (nDCG@20). A KNN classifier is employed to perform the prediction using extracted pre-retrieval query performance predictors as input features. Each query feature set is assigned one of three labels: 0 (BM25), 1 (DFRee), or 2 (equality). Specifically, if the nDCG@20 score of BM25 is higher than that of DFRee for a given query, label 0 is assigned. Conversely, if the DFRee score is higher, label 1 is assigned. In cases where the scores are equal, label 2 is assigned.

D. Results

Table II lists the average nDCG@20 scores of the term weighting models across the query collection datasets. The applied classifier is denoted as PBM, which refers to the prediction-based method. The bold scores indicate the higher scores. The superscripts *B* and *D* indicate that the average retrieval performance of the PBM is statistically significant compared to DFRee and BM25, receptively, at the 0.05 significance level. According to the table results, KNN-based prediction of the term weighting model using query performance predictors produces the highest effectiveness scores. Furthermore, the produced result for CW09B is statistically significantly higher than that produced by DFRee. Also, a similar situation is observed in the CW12B and GOV2 query collection datasets. The produced scores for these datasets are statistically significantly higher than BM25.

Fig. 1 illustrates the per-query comparisons of the PBM against DFRee and BM25 for each dataset. The bars in the upper-right region represent the queries for which PBM achieves higher nDCG@20 scores than the corresponding baseline model. On the other hand, the bars in the lower-left region indicate queries where PBM performs worse. The length of the bar reflects the magnitude of these differences. In the figures, the area covered by the upper-right bars exceeds that of the lower-left bars. It can be inferred from the figures that PBM is accurate in prediction and especially on the queries that have marginal differences between term weighting models.

Fig. 2 also compares the per-query retrieval scores sorted in ascending order for each term weighting model. Each sub-figure corresponds to a specific query collection. Here, the blue, red, and yellow lines indicate the BM25, DFRee, and PBM models. The line for the PBM model usually appears above the other lines of models. This shows that the PBM usually produces more robust results than others, and an increment in average performance is usually distributed across the queries.

III. DISCUSSION AND CONCLUSIONS

This study investigated the use of pre-retrieval query performance predictors to guide the selection of term weighting models in information retrieval systems. The prediction-based method (PBM), leveraging a KNN classification, dynamically selects a term weighting model between BM25 and DFRee for each query. Experimental results on standard TREC datasets (CW09B, CW12B, and GOV2) demonstrated that PBM has achieved higher nDCG@20 scores compared to either baseline model. The improvements have been statistically significant in multiple cases, and this highlights the effectiveness of per-query model selection.

Per-query analyses further revealed that PBM not only improved average retrieval effectiveness but also enhanced robustness across queries. Figures 1 and 2 show that PBM frequently outperformed the baselines, particularly for queries with marginal performance score differences between models, and that performance gains are broadly distributed rather than concentrated on a few queries. These findings indicate that query performance predictors can successfully identify situations in which a specific term weighting model is likely to be more effective. This allows IR systems to mitigate performance variability across queries.

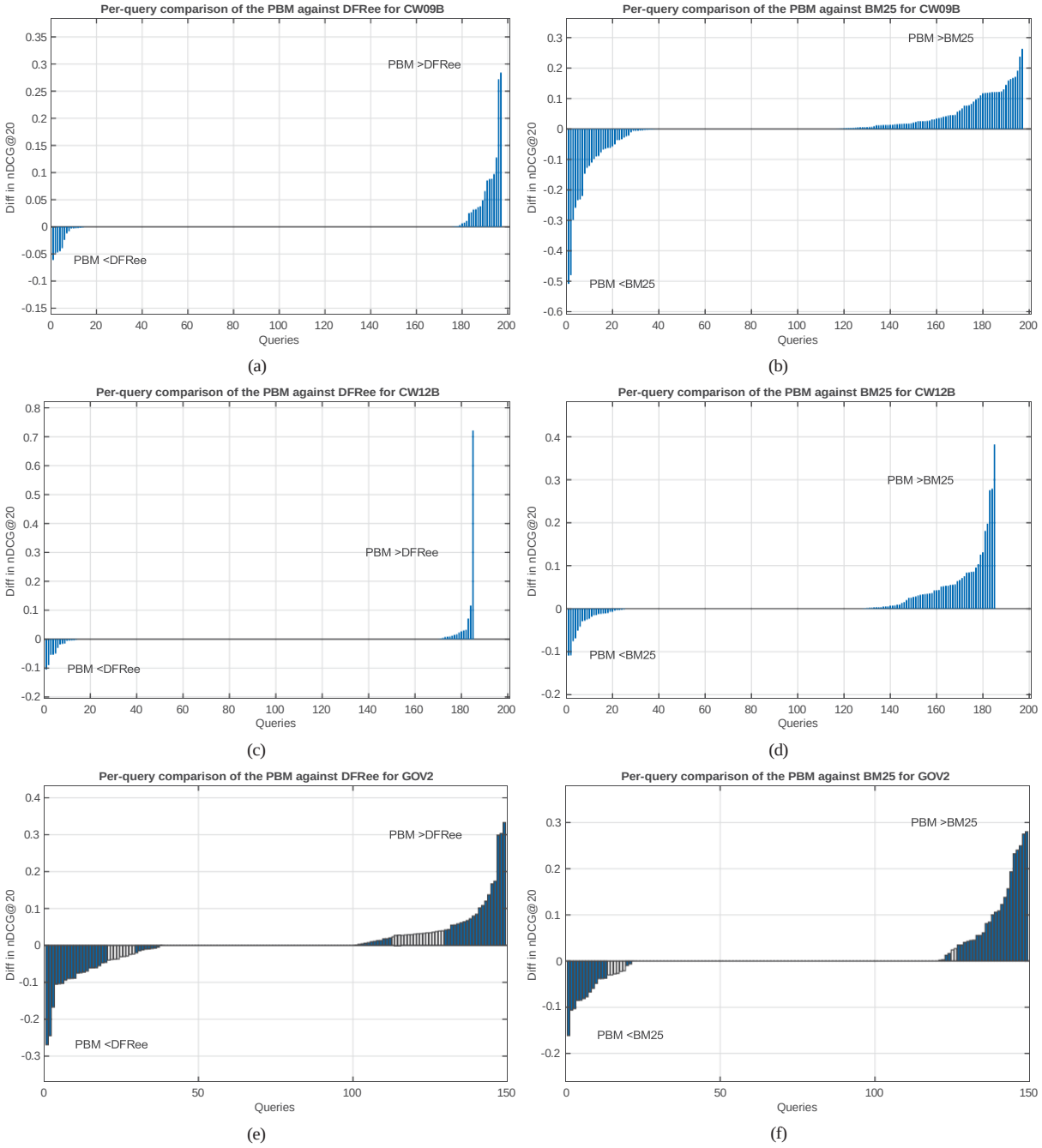

Fig. 1. Visual comparison of the PBM against base term weighting models is presented. The right upper bars represent the number of queries and performance gain with the PBM. The left bottom bars represent the number of queries and performance loss with the PBM.

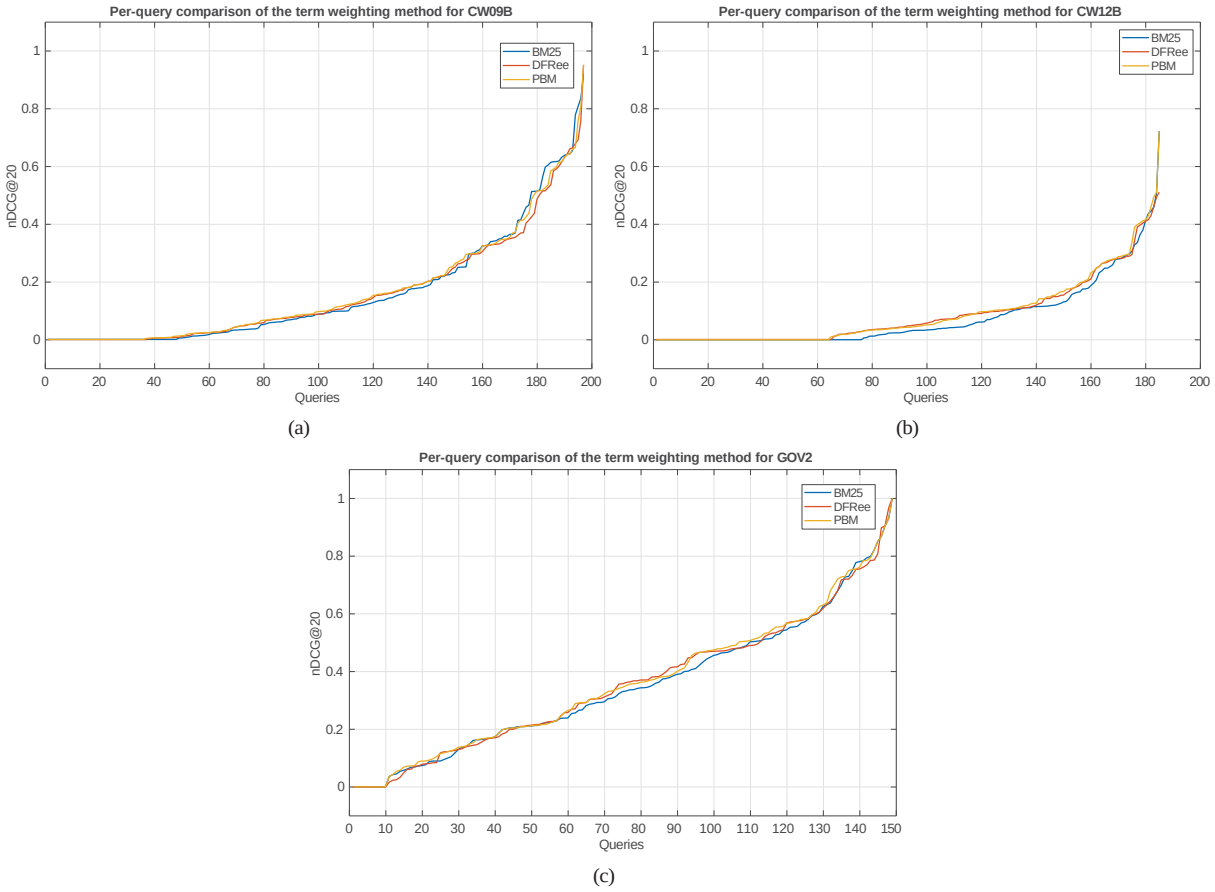

Fig. 2. Per-query performance comparisons of term weighting models and PBM according to CW09B, CW12B, and GOV2.

Despite its effectiveness, this study has several limitations. Only two term weighting models (BM25 and DFRee) are considered, which limits the generalizability of the prediction-based method to other models; however, they are more widely used term weighting models. Although there are other metrics, the experiments focus on nDCG@20 as the evaluation metric, which is a more suitable metric for evaluating web information retrieval.

In conclusion, the prediction-based method demonstrates that pre-retrieval query performance predictors can be effectively used to select term weighting models on a per-query basis. This usually results in significant improvements in both average effectiveness and robustness. These findings provide a promising direction for adaptive IR systems to the specific characteristics of individual queries.

REFERENCES

[1] C. Buckley and E. M. Voorhees, "Retrieval evaluation with incomplete information," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004, pp. 25–32.

[2] G. Go¨ksel, A. Arslan, and B. T. Dinc¸er, "A selective approach to stemming for minimizing the risk of failure in information retrieval systems," *PeerJ Computer Science*, vol. 9, p. e1175, 2023.

[3] A. Aydın, A. Arslan, and B. T. Dinc¸er, "A set of novel html document quality features for web information retrieval: Including applications to learning to rank for information retrieval," *Expert Systems with Applications*, vol. 246, p. 123177, 2024.

[4] A. Arslan and B. T. Dinc¸er, "A selective approach to index term weighting for robust information retrieval based on the frequency distributions of query terms," *Information Retrieval Journal*, vol. 22, no. 6, pp. 543–569, 2019.

[5] C. Buckley, "Why current ir engines fail," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004, pp. 584–585.

[6] M. F. Porter, "An algorithm for suffix stripping," in *Readings in Information Retrieval*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 313–316.

[7] R. Krovetz, "Viewing morphology as an inference process," in *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '93. Pittsburgh, Pennsylvania, USA: ACM, 1993, pp. 191–202.

[8] J. B. Lovins, "Development of a stemming algorithm." *Mech. Transl. Comput. Linguistics*, vol. 11, no. 1-2, pp. 22–31, 1968.

[9] J. H. Paik, D. Pal, and S. K. Parui, "A novel corpus-based stemming algorithm using co-occurrence statistics," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '11. Beijing, China: Association for Computing Machinery, 2011, pp. 863–872.

[10] P. Singh and P. K. Bhowmick, "Neural network guided fast and efficient query-based stemming by predicting term co-occurrence statistics," *SN Computer Science*, vol. 3, no. 3, p. 198, 2022.

[11] F. Peng, N. Ahmed, X. Li, and Y. Lu, "Context sensitive stemming for web search," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*.

ACM, 2007, pp. 639–646.
[12] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, Apr. 2009.
[13] G. Amati, "Divergence from randomness models," in *Encyclopedia of Database Systems*. Springer, 2009, pp. 929–932.
[14] ̇I. Kocabas¸, B. T. Dinc¸er, and B. Karaog˘lan, "A nonparametric term weighting method for information retrieval based on measuring the divergence from independence," *Information retrieval*, vol. 17, no. 2, pp. 153–176, 2014.
[15] K. Ja¨rvelin and J. Keka¨la¨inen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, Oct. 2002.
[16] ——, "IR evaluation methods for retrieving highly relevant documents," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '00, Athens, Greece, 2000, pp. 41–48.
[17] T.-Y. Liu *et al.*, "Learning to rank for information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.
[18] C. Macdonald, R. L. Santos, and I. Ounis, "The whens and hows of learning to rank for web search," *Information Retrieval*, vol. 16, no. 5, pp. 584–628, 2013.
[19] B. van den Akker, I. Markov, and M. de Rijke, "Vitor: learning to rank webpages based on visual features," in *The world wide web conference*, 2019, pp. 3279–3285.
[20] Z. Dai and J. Callan, "Context-aware term weighting for first stage passage retrieval," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1533–1536. [Online]. Available: https://doi.org/10.1145/3397271.3401204
[21] A. Mallia, O. Khattab, T. Suel, and N. Tonellotto, "Learning passage impacts for inverted indexes," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 1723–1727.
[22] J. Lin and X. Ma, "A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques," *arXiv preprint arXiv:2106.14807*, 2021.
[23] T. Formal, B. Piwowarski, and S. Clinchant, "Splade: Sparse lexical and expansion model for first stage ranking," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 2288–2292.
[24] R. Pradeep, R. Nogueira, and J. Lin, "The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models," *arXiv preprint arXiv:2101.05667*, 2021.
[25] H. Zhuang, Z. Qin, R. Jagerman, K. Hui, J. Ma, J. Lu, J. Ni, X. Wang, and M. Bendersky, "Rankt5: Fine-tuning t5 for text ranking with ranking losses," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 2308–2313. [Online]. Available: https://doi.org/10.1145/3539618.3592047
[26] C. L. Clarke, N. Craswell, I. Soboroff *et al.*, "Overview of the trec 2004 terabyte track." in *TREC*, vol. 4, 2004, p. 74.
[27] D. Carmel and E. Yom-Tov, *Estimating the query difficulty for information retrieval*. Morgan & Claypool Publishers, 2010.
[28] B. He and I. Ounis, "Query performance prediction," *Information Systems*, vol. 31, no. 7, pp. 585–594, 2006, (1) SPIRE 2004 (2) Multimedia Databases. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437905000955
[29] C. Hauff, "Predicting the effectiveness of queries and retrieval systems," *SIGIR Forum*, vol. 44, no. 1, p. 88, Aug. 2010. [Online]. Available: https://doi.org/10.1145/1842890.1842906
[30] B. He and I. Ounis, "Inferring query performance using pre-retrieval predictors," in *International symposium on string processing and information retrieval*. Springer, 2004, pp. 43–54.
[31] ̇I. Kocabas¸, B. T. Dinc¸er, and B. Karaog˘lan, "A nonparametric term weighting method for information retrieval based on measuring the divergence from independence," *Information retrieval*, vol. 17, no. 2, pp. 153–176, 2014.
[32] Y. Zhao, F. Scholer, and Y. Tsegay, "Effective pre-retrieval query performance prediction using similarity and variability evidence," in *European conference on information retrieval*. Springer, 2008, pp. 52–64.