A Simple Framework for Active Learning to Rank

Qingzhong Wang **Haifang Li** **Haoyi Xiong** **Wen Wang** **Jiang Bian**
Yu Lu **Shuaiqiang Wang** **Zhicong Cheng** **Dawei Yin**
Dejing Dou
{wangqingzhong, lihaifang, xionghaoyi}@baidu.com
Baidu Inc., Beijing, China

Abstract

Learning to rank (LTR) plays a critical role in search engine—there needs to timely label an extremely large number of queries together with relevant webpages to train and update the online LTR models. To reduce the costs and time consumption of queries/webpages labeling, we study the problem of *Active Learning to Rank* (**active LTR**) that selects unlabeled queries for annotation and training in this work. Specifically, we first investigate the criterion–*Ranking Entropy (RE)* characterizing the entropy of relevant webpages under a query produced by a sequence of online LTR models updated by different checkpoints, using a Query-By-Committee (QBC) method. Then, we explore a new criterion namely *Prediction Variances (PV)* that measures the variance of prediction results for all relevant webpages under a query. Our empirical studies find that RE may favor low-frequency queries from the pool for labeling while PV prioritizing high-frequency queries more. Finally, we combine these two complementary criteria as the sample selection strategies for active learning. Extensive experiments with comparisons to baseline algorithms show that the proposed approach could train LTR models achieving higher Discounted Cumulative Gain (*i.e.*, the relative improvement ΔDCG$_4$=1.38%) with the same budgeted labeling efforts, while the proposed strategies could discover 43% more valid training pairs for effective training.

1 Introduction

Generally, ranking the retrieved contents plays a critical role in a search engine, where learning to rank (LTR) is a standard workhorse. To achieve better ranking performance, we need to use a large amount of annotated data to train a LTR model. However, it is extremely expensive and time-consuming to label the ranks of relevant webpages for every query [1]. To address this issue, active learning [2, 3] to pickup a small number of most informative queries and relevant webpages for labeling is requested.

In this paper, inspired by uncertainty-based active learning methods, we present a simple yet effective approach to active learning for ranking. First, we investigate *Ranking Entropy (RE)*, which characterizes the uncertainty of the ranking for every relevant webpage under a query using a Query-By-Committee (QBC) method [4]. Intuitively, RE could discover queries with ranking uncertainty — *i.e.*, the predicted ranks of webpages in a query are indistinguishable using the LTR model. However RE is also biased in favor of the low-frequency queries, *i.e.*, the queries are less searched by users, as their are no sufficient supervisory signals (*e.g.*, click-throughs) to train LTR models for fine predictions. The bias to the low-frequency queries would not bring sufficient information gain to LTR training. To alleviate this problem, we study yet another criterion – *Prediction Variance (PV)*, which refers to the variances of rank prediction results among all relevant webpages for a query. Intuitively, we assume a query pairing to multiple webpages that have clearly distinguished orders of ranking as a query with *high diversity*. We further assume the variance of rank prediction results would faithfully

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

characterize the variance of ground truth rank labels–*i.e.*, the diversity of webpages in a query. We thus propose to use PV as a surrogate to the diversity of webpages in a query. Please refer to *Section 3* for detailed comparisons and empirical analysis with real data.

More specifically, we report our practices in using above two criteria to design queries selection strategies for active learning. We conducted comprehensive empirical studies on these two criteria using realistic search data. The empirical studies shows that the use of RE results in the bias to the low-frequency queries, while the use of PV leads to the potential over-fittings to the high-frequency queries. When incorporating low-frequency queries (less searched queries) in labeling, the active learner might not be able to train LTR models well, due to the lack of supervisory signals (*e.g.*, click-throughs) to distinguish the webpages for the queries. In contrast, when using high-frequency queries (hot queries) in labeling, the active learner might not be able to adapt the out-of-distribution queries (which is critical for ranking webpages at web-scale). Please see also in *Sections 3.2 and 3.3* for the details of the criterion and empirical observations.

We combine above two complementary criteria as the query selection strategies for active learning. Extensive experiments with comparisons to baseline algorithms show that the proposed approach (*i.e.*, the combination of RE and PV) could train LTR models achieving higher accuracy with fewer webpages labeled. Specifically, we have made contributions as follows.

- We study the problem of active learning for ranking for search engines, where we focus on selecting queries together with relevant webpages for annotations to facilitate LTR models training and updates. To the best of our knowledge, this work is the first to study simple yet effective strategies with sample selection criterion that applies to real search engine data.
- We first consider a commonly-used uncertainty metrics for active learning of LTR, namely *Ranking Entropy (RE)*. We find the use of RE could be biased by the frequency of queries, *i.e.*, low-frequency queries normally have higher RE scores, as LTR models usually have not been well trained to rank webpages in such queries due to the lack of supervisory signals. To de-bias RE, we propose to study yet another diversity-based criteria – *Prediction Variance (PV)* that may favor high-frequency queries and are highly correlated to the true label variance of webpages under the query. In this way, we combine the two criteria for additional performance improvements.
- We conduct extensive experiments, showing that our proposed approach is able to significantly improve the performance of LTR using real data collect form a popular search engine. Specifically, we compare our proposals (the combination of RE and PV) with a wide range of sample selection criteria for active learning, including random pickup, expected loss prediction (aka ELO-DCG) [5]. The comparisons show that our proposals outperform other criteria, which discovers 43% more validate training pairs and improves DCG (*e.g.*, $\Delta \mathrm{DCG}_4$=0.35%$\sim$1.38%) using the same budgeted labeling efforts under fair comparisons.

Note that in this work, we focus on the low-complexity criteria of sample selection in active learning for LTR. There also exists some sample set selection algorithms [6, 7] for active learning in the high-order polynomial or even combinatorial complexity over the number of unlabeled samples, which is out of the scope of this paper as we intend to scale-up active learning of LTR with trillions of unlabeled webpages archived.

2 Related Works

The goal of *active learning* (AL) is to select the most informative samples in the unlabelled data pool for annotation to train a model [8]. Generally, AL models are able to achieve similar performance but use fewer annotated data points. To select the most informative samples for labeling, two categories of methods, *i.e.*, diversity-aware criteria and uncertainty-aware criteria, for sample selection have been studied. The diversity-aware methods [9, 10] measure the diversity of every subset of unlabeled samples and select the sample set with top diversity for labeling, where the core-set selection [11] leveraging core-set distance of intermediate features is a representative method here. While diversity-aware methods work well on the small datasets, they might fail to scale-up over large datasets due to the needs of subset comparisons and selections.

The uncertainty-aware methods [12, 13, 14, 15, 16] screen the pool of unlabeled samples and select samples with top uncertainty in the context of training model (*e.g.*, LTR models here) for

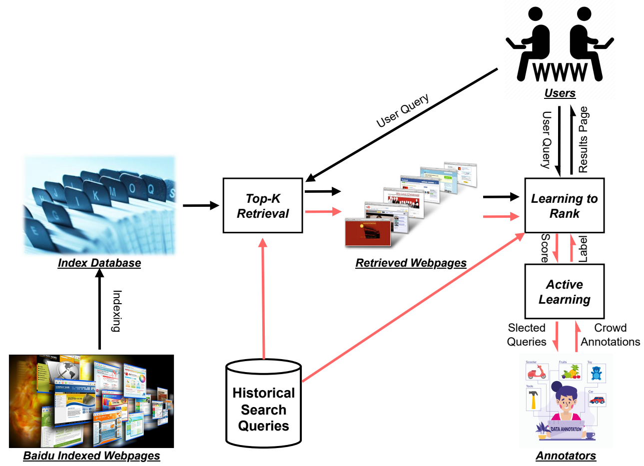

Figure 1: An overview of a typical search engine with the proposed active learning process. While the search engine records every search query from users and stores them in *Historical Search Queries*, it periodically picks up the NEW queries appeared within the last ONE month for annotation and re-trains LTR models with annotated data.

labeling. While uncertainty-aware methods could easily scale-up over the large datasets due to the low complexity, a wide variety of uncertainty criteria have been proposed, such as Monte Carlo estimation of expected error reduction [17], distance to the decision boundary [18, 19], margin between posterior probabilities [20], and entropy of posterior probabilities [21, 22, 23].

The most relevant works to this study are [24, 25, 5, 26]. As early as 2010, Long *et al.* [24, 5] proposed the expected loss optimization (ELO) framework, which selects and labels most informative unlabelled samples for LTR, and incorporate a predictor for discounted cumulative gain (ELO-DCG) to estimate the expected loss of given queries and documents. The work [25] further confirmed ELO with DCG could work well with any rankers at scale and deliver robust performance. Cai *et al.* [26] followed the settings of ELO and extended DCG through incorporating the kernel density of queries, so as to balance sample distribution and the model-agnostic uncertainty for sample selection.

3 Active Learning to Rank

In this section, we first review the system design of *active learning to rank* (Active LTR) for web search, then present our proposed selection criteria for active learning with empirical observations.

As shown in Figure 1, given a search query, denoted as q, from a user, the search engine frequently first retrieves all relevance webpages, denoted as $\{w_1, w_2, \dots\}$, from the dataset and sorts the top-K relevant webpages for the best user reading experience through ranking. To rank every webpage under the query, the search engine pairs every webpage with the query to formed a query-webpage pair, e.g., (q, w), and then extracts features from (q, w), denoted as the feature vector (x_q, x_w), where x_q denotes query-relevant features and x_w denotes webpage-relevant features, and adopts the *learning to rank* (LTR) model to predict the ranking score, e.g., {**bad, fair, good, excellent, perfect**}[1] using (x_q, x_w).

To train the LTR model, the search engine usually collects the *Historical Search Queries* $\mathcal{Q} = \{q_1, q_2, \cdots\}$ and archives relevant webpages $\mathcal{W} = \{w_{1,1}, w_{1,2}, \cdots\}$, where $w_{i,j}$ denotes the jth webpage associates with q_i. To scale-up the *active LTR* on trillions of webpages/queries while ensuring the timeness of search engine, our active learning system (red path in Figure 1) periodically picks up NEW queries appeared within the last ONE month i.e., S $\subset \mathcal{Q}$, pairs every selected query in S with retrieved webpages and extracts feature vectors to form the unlabeled datasets denoted as $\mathcal{T} = \{(x_{q_1}, x_{w_{1,1}}), (x_{q_1}, x_{w_{1,2}}), \cdots\}$. Finally, the search engine recruits annotators to label $\mathcal{T}$ and retrains the LTR model with annotated data.

[1]For human annotations, labels 0, 1, 2, 3, 4, 5 denote bad, fair, good, excellent and perfect, respectively.

3.1 Sample Selection Criteria for Active LTR

In this section, we present the two criteria proposed for active learning to rank webpages.

3.1.1 Ranking Entropy (RE)

Uncertainty is one of the most popular criteria in active learning and *Query-By-Committee* (QBC) [4] approach has been widely applied to estimate the uncertainty scores of the unlabelled data. In this paper, we apply QBC to compute the *Ranking Entropy* (RE) of each webpage. Normally, there are M models $\{h_m(x_q, x_w)|m = 1, \cdots, M\}$ to constitute a committee. Given the representation of a query-webpage pair $(x_{q_i}, x_{w_{i,j}}) \in \mathcal{T}$, the committee would provide a set of scores $\mathcal{S}_{i,j} = \{h_m(x_{q_i}, x_{w_{i,j}})|m = 1, \cdots, M\}$. Then, for any two webpages $\{w_{i,u}, w_{i,v}\}$ associate to q_i, we can easily calculate the probability that webpages $w_{i,u}$ is ranked higher than $w_{i,v}$ under query q_i, denoted as the probability of $w_{i,u} \succ w_{i,v}$, *i.e.*,

$$\pi_i^m(w_{i,u} \succ w_{i,v}) = \frac{1}{1 + exp\left(\frac{-h_m(x_{q_i}, x_{w_{i,u}}) + h_m(x_{q_i}, x_{w_{i,v}})}{T}\right)}, \tag{1}$$

where T denotes the temperature and $w_{i,u} \succ w_{i,v}$ denotes that $w_{i,u}$ is more relevant than $w_{i,v}$ under query q_i.

Similar to SoftRank [27], we can obtain the distribution over ranks based on the probabilities calculated by Eq. (1). We define the initial rank distribution of the vth webpage $w_{i,v}$ as $p_{i,v}^{1,m}(r) = \delta(r)$, where $\delta(r) = 1$ if $r = 0$ and 0 otherwise, then we can calculate the distribution in the kth step as follows:

$$\begin{aligned} p_{i,v}^{k,m}(r) =& p_{i,v}^{k-1,m}(r-1)\pi_i^m(w_{i,u} \succ w_{i,v}) + \\ & p_{i,v}^{k-1,m}(r)\left(1 - \pi_i^m(w_{i,u} \succ w_{i,v})\right), \end{aligned} \tag{2}$$

and the distribution in the last step will be the final ranking distribution. The computing procedure is shown in Alg. 1, where one webpage is added to the list for comparison in each step and the ranking distribution is updated using Eq. (1).

For each webpage $w_{i,j}$, we can obtain a set of ranking distributions $\{p_{i,j}^m(r)|m = 1, \cdots, M; r = 1, \cdots, N_i\}$ using Alg. 1, where N_i denotes the number of webpages associate with q_i. Finally, for every query-webpage pair with the feature vector $(x_{q_i}, x_{w_{i,j}}) \in \mathcal{T}$, we use the average distribution over the committee to compute its entropy score as follow

$$p_{i,j}(r) = \frac{1}{M}\sum_{m=1}^{M} p_{i,j}^m(r), \tag{3}$$

$$E_{i,j} = -\sum_{r=1}^{N_i} p_{i,j}(r)\log_2 p_{i,j}(r). \tag{4}$$

Note that the goal of this paper is to select queries, hence, for a query q_i, we employ the average entropy of the webpages $\{w_{i,j}|j = 1, \cdots, N_i\}$ that associate with q_i, *i.e.*,

$$RE(q_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} E_{i,j}. \tag{5}$$

Higher $RE(q_i)$ refers to larger uncertainty in ranking results across the LTR models in the committee. Active learners are expected to pickup queries with large *Ranking entropy*, i.e., higher $RE(q_i)$ for $q_i \in \mathcal{Q}$, for annotation and training.

3.1.2 Prediction Variance (PV)

In our work, we assume a query, pairing to multiple webpages that are with clearly distinguished orders of ranking, as a *query with high diversity*. While the true orders of ranking could be obtained through human annotations (*i.e.*, labeling every webpage under the query using scores of five levels 0, 1, 2, 3, and 4 in this work), we propose to use the rank prediction results of a trained LTR model to measure such diversity. Given a checkpoint of the online GBRank [28] model, we propose to adopt

Algorithm 1: Ranking Distribution

```
# Pseudo code in Python
Data: {π_i^m(w_{i,u} ≻ w_{i,v}) | u, v = 1, ..., N_i; u ≠ v}
Result: {p_{i,v}^m | v = 1, ..., N_i}
for v in range(N_i) do
    p_{i,v}^m = [1, 0, ..., 0] # N_i elements
    p_tmp = [0, ..., 0] # N_i elements
    π_v = [π_i^m(w_{i,v} ≻ w_{i,1}), ..., π_i^m(w_{i,v} ≻ w_{i,N_i})]
    for u in range(1, N_i) do
        for r in range(v + 1) do
            if r==0 then
                α = 0
            else
                α = p_{i,v}^m[r − 1]
            end
            p_tmp[r] = p_{i,v}^m[r] * π_v[u − 1] + α * (1 − π_v[u − 1])
        end
        p_{i,v}^m = p_tmp
    end
end
```

the variance of predicted ranking scores (namely Prediction Variance (PV)to measure the *diversity of webpages* for the query.

Similar to RE, we also use the predictions of the committee to compute the prediction variance. Given the outputs of the committee $\mathcal{S}_{i,j} = \{h_m(x_{q_i}, x_{w_{i,j}}) | m = 1, \cdots, M\}$, the prediction variance of a query q_i with N_i retrieved webpages can be computed using the following equations:

$$\mu_m(q_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} h_m\left(x_{q_i}, x_{w_{i,j}}\right), \tag{6}$$

$$STD_m(q_i) = \sqrt{\frac{1}{N_i} \sum_{j \in 1}^{N_i} \left(h_m\left(x_{q_i}, x_{w_{i,j}}\right) - \mu_m(q_i)\right)^2}, \tag{7}$$

Finally, we calculate the prediction variance $PV(q_i)$ as follows:

$$PV(q_i) = \frac{1}{M} \sum_{m=1}^{M} STD_m(q_i). \tag{8}$$

For active learning, we assume queries with large prediction variances, i.e., higher $PV(q_i)$ for $q_i \in \mathcal{Q}$, as the candidates for annotation and training.

To be simple, we use the weighted sum of RE and PV as the acquisition function in active learning. For each query $q_i \in \mathcal{Q}$, the acquisition function is

$$f(q_i) = RE(q_i) + \alpha * PV(q_i), \tag{9}$$

where α is a hyper-parameter to balance ranking uncertainty and webpage variance. We select the queries that have the largest values of $f(q_i)$ in each cycle of active learning.

4 Experiments

In this section, we present the results of experiments compared with baseline algorithms.

4.1 Setups

First, we collect 15,000 queries from a popular search engine and each query has 60 webpages on average. we split the dataset (15,000 queries) into training set (14,000 queries) and validation set

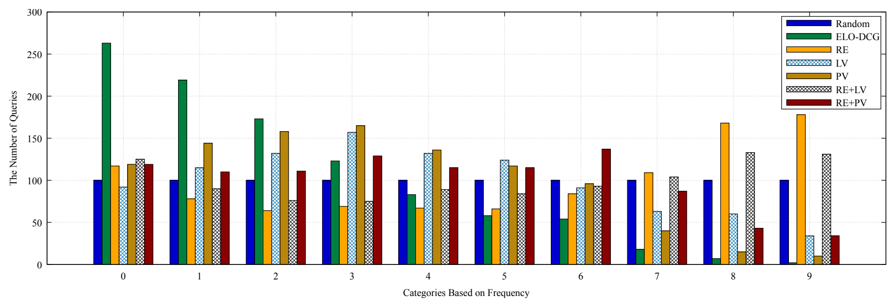

Figure 2: The distribution of 1,000 selected queries using different criteria. LV stands for label variance, PV for prediction variance and RE for ranking entropy.

(1,000 queries). In the beginning of active learning, we randomly select N_0 queries from the training set as the base and in each cycle of active learning, we set the batch size $bs = 100$, *i.e.*, we select 100 queries from the pool (the rest of the training set) using the acquisition function. The quota is 2,000 queries, *i.e.*, we run active learning for 20 cycles. We also conduct ablation studies on the value of α in Eq. (9), where $\alpha = \{0.5, 1.0, 1.5\}$. We set the number of committee M to 9, *i.e.*, 9 variants of GBRank with different numbers of trees (100, 300, 500) and maximum depth (1, 3, 5).

To evaluate the performance of a LTR model, we use *Discounted Cumulative Gain* (DCG), computing as follows:

$$DCG_K = \sum_{k=1}^{K} \frac{G_k}{\log_2(k+1)}, \tag{10}$$

where G_k denotes the weight assigned to the webpage's label at position k. A higher G_k indicates that the webpage is more relevant to the query. Also, a higher DCG_K indicates a better LTR model. In this paper, we consider the DCG of top 4 ranking results, *i.e.*, DCG_4. In addition, we consider another important metric – the percentage of the irrelevant webpages in top K, which is computed as follows:

$$R_{01} = \frac{N_{01}}{K}, \tag{11}$$

where N_{01} denotes the number of the irrelevant webpages[2]. Obviously, a lower R_{01} indicates a better LTR model. Also, we consider R_{01} in top 4 in this paper.

There are two baselines for comparison, the first one random selection and the second one is ELO-DCG[5] – an uncertainty-based active learning method for ranking.

4.1.1 Results

Here, we first present the statistical characteristics of selected queries (with webpages retrieved) for annotations, then introduce the details about the valid query-webpage pairs formed from annotation results for training. Finally, we present the performance improvements of proposed criteria in comparisons to baseline criteria, such as ELO-DCG [24, 5].

Fig. 2 shows the distribution over categories of 1,000 selected queries. Category 0 is composed of the most frequent queries, while category 9 contains the least frequent (only one time in the one-month search log) queries. For random selection, we randomly select 100 queries from each category. Compared with random selection, *Label Variance* (LV) prefers relatively frequent queries, such as categories 2-4, but selects fewer low-frequency queries in categories 7-9, indicating that the webpages associate with low-frequency queries have similar human annotations. PV performs similar in high-frequency queries but selects much fewer low-frequency queries. By contrast, RE selects the most low-frequency queries. However, it is difficult to construct enough training pairs if there are too many low-frequency queries, since irrelevant webpages dominates these queries. Interestingly, the existing work – ELO-DCG [24, 5] is in favor of high-frequency queries. Generally, the search engine can well handle high-frequency queries to satisfy user's demands and selecting more high-frequency queries cannot benefit the gain of search.

[2] We consider the webpages with labels of 0 and 1 as irrelevant webpages.

Criterion	# valid pairs	# neg-pos pairs
Random	764,527	534,500
ELO-DCG [5]	959,228 (25%↑)	598,555 (12%↑)
LV	1,039,474 (36%↑)	757492 (42%↑)
PV	979,051 (28%↑)	704,621 (32%↑)
RE	823,411 (8%↑)	562,464 (5%↑)
Ours (RE+PV)	**1,091,176 (43%↑)**	**803,723 (50%↑)**
RE+LV(upper bound)	1,149,083 (50%↑)	827,133 (55%↑)

Table 1: The number of training pairs obtained by using different criteria and the relative improvement compared with random selection. If two webpages associate with a query have different labels, then they constitute a valid pair. Neg-pos pair denotes that a pair is composed of an irrelevant and a relevant webpages. We use each approach to select 1,000 queries to obtain the statistics.

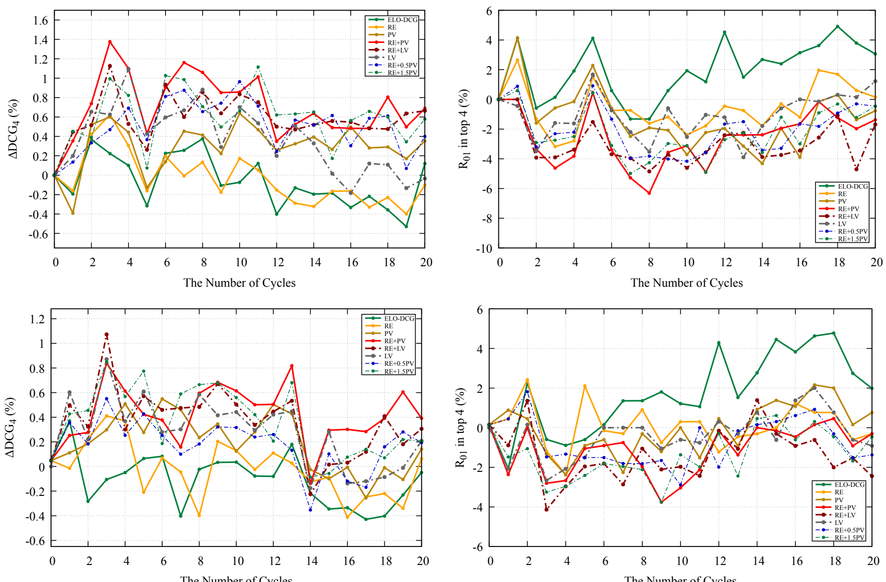

Figure 3: The relative improvements of DCG_4 (*i.e.*, ΔDCG_4) and R_{01} (*i.e.*, ΔR_{01}) compared with using random selection in each active learning cycle with the same budget. Top: the base set is composed of 100 queries. Bottom: the base set is composed of 500 queries.

For GBRank [28] that uses pairwise loss, the number of training pairs is crucial. With more training pairs fed to the training procedure, LTR models are expected to deliver better performance. Table 1 presents the number of pairs obtained using different approaches. We can easily conclude that the proposed approach is able to obtain more training pairs compared with random selection and the existing work – ELO-DCG [24, 5]. In terms of the number of valid pairs composed of two webpages with different human annotated labels, random selection obtains 764,527 pairs for 1,000 queries, while the number increases by 25% when using ELO-DCG [24, 5]. Using the criteria LV and PV that improve the diversity among webpages, the number of valid pairs can be improved by 36% and 28%, respectively. Only using RE can also achieve a 8% increase compared with random selection, but it is inferior to ELO-DCG, LV and PV since RE selects more low-frequency queries and most webpages associate with these queries are with label 0 (see Fig. 2). Though selecting more low-frequency queries could benefit solving the problem of unusual queries and attracting more users, it is difficult to retrieve relevant webpages for these queries and irrelevant webpages are less useful to train GBRank. Combining RE and PV is able to alleviate the problem. The number of valid pairs surges by 43% and 50% by using RE+PV and RE+LV respectively, which is a remarkable improvement. Considering the number of neg-pos pairs, our proposed approach is also superior to random selection and ELO-DCG [24, 5]. Generally, the number of neg-pos pairs is related to the percentage of the irrelevant webpages in top K since using more neg-pos pairs to train a LTR model, it would be easier to distinguish relevant webpages from irrelevant webpages. Using RE+PV the number of neg-pos pairs dramatically raises by 50% compared with random selection and it is also boosted by 34% compared to the existing active learning approach – ELO-DCG [24, 5].

LTR Performance Comparisons In this experiments, we use the valid query-webpage pairs obtained by various strategies to train LTR models (GBRank models with cross-entropy loss) and compare the ranking quality of these LTR models on our validation dataset of 1,000 queries. The ranking quality is measured using DCG. Fig. 3 shows the comparison among different approaches.

Let's first pay attention to base100 – the top two sub-figures in Fig. 3. The proposed approach RE+PV achieves better performance than random selection and ELO-DCG [24, 5]. We can see that using RE+PV selected queries to train GBRanK, DCG_4 raises faster than its counterparts, such as random selection, ELO-DCG and RE. Compared with random selection, the relative improvement of DCG_4 ranges from 0.35% to 1.38% using RE+PV. And compared to the existing work ELO-DCG [24, 5], the proposed RE+PV boosts DCG_4 by at least 0.37%. While random selection outperforms ELO-DCG and RE when selecting more training data. The possible reason is that ELO-DCG and RE are biased by query frequency, *e.g.*, ELO-DCG prefers high-frequency queries, whereas RE is in favor of low-frequency queries. Interestingly, ELO-DCG and RE performs similar to each other, though the distributions of the selected queries are different. Looking at PV and LV that are related to the diversity among webpages, both outperform random selection in most cycles. When it comes to R_{01} (top-right sub-figure in Fig. 3), the proposed RE+PV also achieves competitive performance compared with its counterparts. *E.g.*, R_{01} drops by at most 6.31% compared to random selection. Although ELO-DCG is able to obtain higher DCG_4 scores at the beginning of AL cycles, it performs even worse considering the metric R_{01}. The reason is that ELO-DCG selects more webpages with label 2 but less webpages with label 0, hence, it is difficult for GBRank to distinguish irrelevant webpages.

Moving on to base500 (bottom sub-figures in Fig. 3), our proposed approach – RE+PV also achieves higher DCG_4 than random selection and ELO-DCG [24, 5]. The relative improvement of DCG_4 is at most 0.84% compared to random selection. Moreover, R_{01} shows decreases in most cycles, *e.g.*, in cycle 9, R_{01} drops by 3.76% using RE+PV.

5 Conclusion

In this work, we revisited the problem of active learning for ranking webpages, where the key problem is to establish the training datasets for learning to rank (LTR) models. Given trillions of queries and relevant webpages retrieved for every query, the goal of active learning is to select a batch of queries for labeling and trains the current LTR model with the newly-labeled datasets incrementally, where the labels here refer to the ranking score of every webpage under the query. To achieve the goals, for every query, this work proposed two new criteria–*Ranking Entropy (RE)* and *Prediction Variances (PV)* that could measure the *uncertainty of the current LTR model to rank webpages in a query* and the *diversity of ranking scores for webpages in a query* respectively. Specifically, RE estimates the entropy of relevant webpages under a query produced by a sequence of online LTR models updated by different checkpoints, using a Query-By-Committee (QBC) method, while PV estimates the variance of prediction results for all relevant webpages under a query. Our experimental observations find that RE may pop low-frequency queries from the pool for labeling while PV prioritizing high-frequency queries more. Further, the estimate of PV significantly correlates to the diversity of true ranking scores of webpages (annotated by human) under a query and correlates to the information gain of LTR. Finally, we combine these two complementary criteria as the sample selection strategies for active learning. Extensive experiments with comparisons to baseline algorithms show that the proposed approach could train LTR models achieving higher DCG (*i.e.*, ΔDCG_4=0.35%$\sim$1.38%) using the same budgeted labeling efforts, while the proposed strategies could discover 43% more valid training pairs for effective training.

References

[1] Burr Settles. From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pages 1–18. JMLR Workshop and Conference Proceedings, 2011.

[2] Burr Settles. Active learning literature survey. 2009.

[3] Siyu Huang, Tianyang Wang, Haoyi Xiong, Jun Huan, and Dejing Dou. Semi-supervised active learning with temporal output discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3447–3456, 2021.

[4] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2):133–168, 1997.

[5] Bo Long, Jiang Bian, Olivier Chapelle, Ya Zhang, Yoshiyuki Inagaki, and Yi Chang. Active learning for ranking through expected loss optimization. *IEEE transactions on knowledge and data engineering*, 27(5):1180–1191, 2014.

[6] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963. PMLR, 2015.

[7] Steve Hanneke and Liu Yang. Minimax analysis of active learning. *J. Mach. Learn. Res.*, 16(12):3487–3602, 2015.

[8] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *JAIR*, 4:129–145, 1996.

[9] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *ICML*, page 79, 2004.

[10] Yuhong Guo. Active instance sampling via matrix partition. In *NIPS*, pages 802–810, 2010.

[11] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

[12] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, pages 1–8, 2007.

[13] Zheng Wang and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *TKDD*, 9(3):1–23, 2015.

[14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, pages 1050–1059, 2016.

[15] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

[16] Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. Uncertainty-guided continual learning with bayesian neural networks. In *ICLR*, 2020.

[17] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. pages 441–448, 2001.

[18] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *JMLR*, 2(Nov):45–66, 2001.

[19] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *ICML*, pages 59–66, 2003.

[20] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *ECML*, pages 413–424, 2006.

[21] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP*, pages 1070–1079, 2008.

[22] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, pages 2372–2379, 2009.

[23] Wenjie Luo, Alex Schwing, and Raquel Urtasun. Latent structured active learning. In *NIPS*, pages 728–736, 2013.

[24] Bo Long, Olivier Chapelle, Ya Zhang, Yi Chang, Zhaohui Zheng, and Belle Tseng. Active learning for ranking through expected loss optimization. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274, 2010.

[25] Mustafa Bilgic and Paul N Bennett. Active query selection for learning rankers. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1033–1034, 2012.

[26] Wenbin Cai, Muhan Zhang, and Ya Zhang. Active learning for ranking with sample density. *Information Retrieval Journal*, 18(2):123–144, 2015.

[27] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 77–86, 2008.

[28] Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 287–294, 2007.