Selective Query processing: A Risk Sensitive Approach - Abstract

Josiane Mothe[1,†], Md Zia Ullah[2,‡]

[1]INSPE, UT2J, Univ. de Toulouse, IRIT UMR5505 CNRS, Toulouse, France
[2]Univ. de Toulouse, IRIT UMR5505 CNRS, Toulouse, France

Abstract

While search engines apply a single optimised search strategy to any user query, selective search, like selective query expansion, aims to apply an adapted search strategy to each query. Search phases include query expansion, search-weighting model, and document ranking. A search strategy is defined by the combination of components and their hyperparameters in these phases. The number of possible search strategies is huge. In this paper, we describe a risk-sensitive approach to optimise the set of search strategies that should be included in a selective search approach. It solves the problem of which and how many search strategies to include in the system. We found that using 20 search strategies is an appropriate trade-off between effectiveness and system complexity. Significant effectiveness improvement is about 23% when compared to L2R documents and about 10% when compared to other selective approaches. This paper is an extended abstract of our paper at CIKM 2021 [1].

Keywords

Information Systems, Information Retrieval, Models and ranking, System effectiveness, Adaptive information retrieval, Query processing, Per-query processing, Query driven parameterisation, Search engine parameters, System selection, Risk sensitive systems

A search engine performs several distinct component processes to answer a query. These include automatic query reformulation, search weighting to decide which documents to retrieve, and ranking retrieved documents. Current practice is to decide on the components and their hyperparameters experimentally: system effectiveness is maximised based on past searches or training queries. Once optimised, the same system is then used for all future queries. This ensures the best performance on average for the training queries, but not for individual queries.

Selective search aims to improve the performance of search engines for individual future queries. For example, *selective query expansion* (SQE) applies query expansion only to those queries that will benefit from it. SQE considers two search strategies, one with automatic query reformulation and one without. With the *selective search* in place, the system can choose among many possible search strategies to process a given query. It can thus be seen as a generalisation of SQE.

Selective query processing (also referred to as selective search) is comprised of two main parts:

[1]J. Mothe, M. Z. Ullah, Defining an optimal configuration set for selective search strategy-a risk-sensitive approach, in: CIKM, 2021, pp. 1335–1345

CIRCLE (Joint Conference of the Information Retrieval Communities in Europe), July 04–07, 2022, Samatan, Gers, France
Josiane.Mothe@irit.fr (J. Mothe); zuacsea@gmail.com (M. Z. Ullah)
0000-0001-9273-2193 (J. Mothe); 0000-0002-4022-7344 (M. Z. Ullah)

- *Search strategy pool*: The system has a pool of search strategies it can choose from;
- *Selective search strategy*: The system selects the search strategy from the pool to be applied to the current query using the “best fit" principle.

In related work, either the search strategy pool is limited from 2 to 8 search strategies, or it contains from 20,000 to 80,000 search strategies. In the latter case, to be applicable in real world systems, the pool needs to be restricted.

The main purpose of this paper is to limit the number of search strategies for the most appropriate ones. The risk sensitive approach we developed in this paper aims to define the search strategies and their number to be considered in a selective search approach. This risk sensitive approach is grounded on Wang *et al.*'s F_{Risk} measure for learning to rank documents [2]. F_{Risk} purpose is to decide on the document ranking for a given query. The risk here is defined as, “The risk [for the system] of performing a given particular query less effectively than a given baseline system" [3].

We have adapted F_{Risk} and defined $Eff_{Risk}(c_j)$ as a function for selecting candidate search strategies. This function measures the risk associated with selecting the search strategy c_j for a given query rather than a reference search strategy c_r. The risk relates to c_r being greater than c_j.

The risk function $Eff_{Risk}(c_j)$ accumulates the risk relative to queries in terms of effectiveness for the training query set. $Eff_{Risk}(c_j)$ is defined in Eq. (1) where $\mathcal{Q}_T$ is the training query set consisting of $\mathcal{T}$ queries, $p(c_r, q)$ is the performance (effectiveness) of the reference search strategy c_r for the query q, and c_j is a search strategy from the initial pool. $Eff_{Risk}(c_j)$ hence accumulates the loss in effectiveness in relation to queries when the search strategy c_j is selected, whereas the reference search strategy c_r would have been the better choice: it corresponds to the maximum possible risk.

$$Eff_{Risk}(c_j) = \frac{1}{T} \sum_{q_i \in \mathcal{Q}_T} \max(0, p(c_r, q_i) - p(c_j, q_i)) \tag{1}$$

In addition to the risk function, we have defined the corresponding reward function. It is based on the potential increase in overall effectiveness (Eq. 2) using the search strategy c_j. The reward function is defined as follows:

$$Eff_{Reward}(c_j) = \frac{1}{T} \sum_{q_i \in \mathcal{Q}_T} \max(0, p(c_j, q_i) - p(c_r, q_i)) \tag{2}$$

The reward function aggregates the improvement in effectiveness that would happen if the system only used c_j for the queries, and c_j performs better than the reference search strategy c_r.

We adapted formulas of Eqs. 1 and 2 to fit the problem of selecting a set of search strategies to keep.

[2] L. Wang, P. N. Bennett, K. Collins-Thompson, Robust ranking models via risk-sensitive optimization, in: SIGIR, 2012, pp. 761–770

[3] B. T. Dinçer, C. Macdonald, I. Ounis, Risk-sensitive evaluation and learning to rank using multiple baselines, in: SIGIR, 2016, pp. 483–492

Let $\mathcal{R}$ be the entire initial search strategy pool and S_{k-1} be the set of search strategies that have already been selected at step k with an initial $S_0 = \{c_r\}$, which is a point of reference or the first selected search strategy. $\mathcal{Q}_T$ is the set of training queries and $p(c_k, q_i)$ denotes the retrieval effectiveness (e.g., nDCG@10) for the query $q_i \in \mathcal{Q}_T$ processed by the search strategy c_k.

Given $\mathcal{R}$ and S_{k-1}, we define the risk for selecting the new search strategy $c_k \in \mathcal{R} \setminus S_{k-1}$ to be added to S_{k-1} at step k using Eq. 1 in terms of effectiveness as follows:

$$E_{RISK}(c_k, S_{k-1}) = \frac{1}{|T|} \sum_{q_i \in \mathcal{Q}_T} \max(0, \max_{c_j \in S_{k-1}} (p(c_j, q_i)) - p(c_k, q_i)) \tag{3}$$

where $\max_{c_j \in S_{k-1}} p(c_j, q_i)$ is the maximum effectiveness for the query q_i in relation to the set of search strategies that have already been selected in S_{k-1}. In Eq. 3, the risk of adding the search strategy c_k is measured as the cumulative decrease in effectiveness which the meta-system can achieve if it chooses c_k rather than the best search strategy in S_{k-1} for each of the training queries: it therefore adapts Eq. 1.

Likewise, we define the reward function using Eq. 2 in terms of effectiveness as follows:

$$E_{REWARD}(c_k, S_{k-1}) = \frac{1}{|T|} \sum_{q_i \in \mathcal{Q}_T} \max(0, p(c_k, q_i) - \max_{c_j \in S_{k-1}} p(c_j, q_i)) \tag{4}$$

The overall gain for the search strategy $c_k \in \mathcal{R} \setminus S_{k-1}$ in relation to the set of training queries and already selected search strategies S_{k-1} is defined as:

$$Gain(c_k, S_{k-1}) = Reward(c_k, S_{k-1}) - (1 + \alpha) Risk(c_k, S_{k-1}) \tag{5}$$

where the functions $Reward(c_k, S_{k-1})$ and $Risk(c_k, S_{k-1})$ refer to the effectiveness-based Eqs. 3 and 4, respectively. The $\alpha \geq 0$ is a risk sensitive parameter that controls the trade-off between risk and reward. In our case, we set α as 0 to weight risk and reward equally. We keep a statistical analysis of this risk sensitive parameter for future work.

Finally, at step k we select the search strategy c_k^* which maximises the overall gain according to the following equation:

$$c_k^* = \underset{c_k \in \mathcal{R} \setminus S_{k-1}}{\operatorname{argmax}} \Big(Gain(c_k, S_{k-1}) \Big) \tag{6}$$

We then update S_{k-1} as follows:

$$S_k = S_{k-1} \cup \{c_k^*\} \tag{7}$$

where S_k is the set of k risk sensitive search strategies selected for a set of training queries $\mathcal{Q}_T$.

The risk sensitive criteria model we propose is generic enough to be applied to any selective search strategy approach.

For the selective search strategy part, we use learning to rank (L2R) algorithms to rank the search strategies as suggested in Deveaud *et al.* [4]. The principle is thus to train a ranking model

[4] R. Deveaud, J. Mothe, M. Z. Ullah, J.-Y. Nie, Learning to adaptively rank document retrieval system configurations, TOIS 37 (2018)

$r(q_i, c_j) = r(f_{i,j})$ to assign a score to a given query-search strategy pair (q_i, c_j) i.e., a given feature vector $f_{i,j}$. More generally, the ranking model can rank all the search strategies for a given query q_i. In this case, the ranking model is $R(q_i, \mathcal{S}) = R(\mathbf{f_i})$, where $\mathcal{S}$ is the set of search strategies and $\mathbf{f}_i = (f_{i,1}, f_{i,2}, \cdots, f_{i,|\mathcal{S}|})$ is the set of feature vectors for the query-search strategy pairs. Like the L2R documents model, the ranking model $R(q_i, \mathcal{S})$ is learned from the training data by minimising the loss function $\mathcal{L}(r; \mathbf{f}, \mathcal{S})$.

To evaluate our contributions, we considered three standard TREC collections from the Adhoc tasks. TREC78 (about 500K newspaper articles), WT10G (1.6 million Web/blog page documents), and GOV2 (25 million web pages).

Search strategies were built by varying the IR components and some of their hyperparameters. We considered the term weighting model and automatic query expansion components from which we considered different variants from the literature. We also considered different values of the hyperparameter related to query expansion. This results in more than $20,000$ search strategies which were the input for the E_{RISK} selection model used to reduce this set based on training queries.

Query-search strategy training examples follow a vector-based representation: the features $f_{i,j}$ depend on the query (q_i), the search strategy (c_j), and a label. We opted for LETOR features that have been used successfully for document ranking models. We calculated LETOR features directly from an initial search that we performed using a reference system (BM25). Finally, the query-search strategy vectors were labelled by the effectiveness of the search strategy when treating that query.

We made k, the number of search strategies, vary and found that 20 is appropriate for real world environments (see Figure 1).

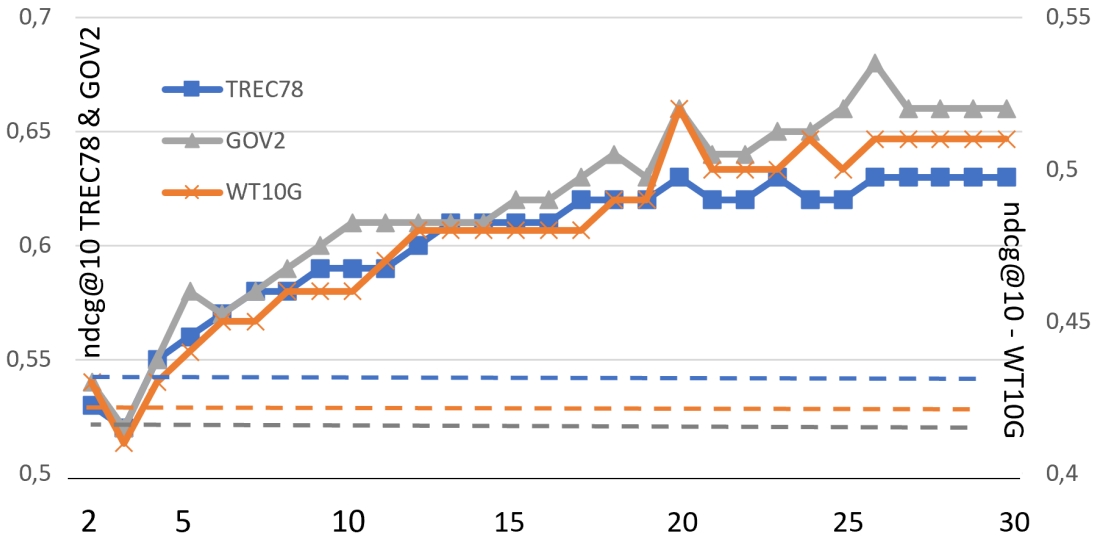

Figure 1: The more search strategies, the better the performance. Ndcg@10 (Y-axis) for different numbers of candidate search strategies (X-axis)- 3 TREC collections. Dotted dash horizontal lines are the single best search strategies.

Table 1 shows additional results on GOV2 TREC adhoc collections. The first block shows

baselines that use a single search strategy for all queries; the second block shows trained selective search strategy (SelSS) including ours ($k = 20$ search strategies); the latest shows oracles. The first oracle is the Best conf. row where the best configuration of the pool is selected *a posteriori*. The second line in this block is when for each query we select the best configuration for that specific query; this is also an *a posteriori* selection. Effectiveness in absolute value, averaged on 3 draws plus standard deviation in square brackets. The best values (excluding Oracle) are in bold font. $\triangle$ (resp. $\uparrow$) indicates statistically significant improvement compared to the L2R documents (resp. Deveaud *et al.*), two-tailed paired t-test ($p < 0.05$). Results on the two other TREC collections were consistent.

From these results, we can see that our method is better than any single query processing strategy (i.e. system configuration), even if we could select it automatically (which is not the case in real life). Our strategy is also better than selective query expansion, where the query processing differ from one query to the other but with just two possible choices. It is also slightly and statistically better than when using 20 000 configurations in the initial pool. This is the most interesting and original results since it shows that we need a certain number of query processing strategies to be effective but not necessarily a huge number which makes the approach more realistic and feasible in real IR systems.

Table 1
Risk-RF outperforms any baseline on all measures

	Methods	MAP	nDCG@10	P@10
Baselines	BM25	.27	.46	.54
	L2R-D SVM^r	.28 [.001]	.49 [.002]	.57 [.003]
	GS	.35 [.005]	.52 [.003]	.62 [.008]
	Best trained	.35 [.005]	.49 [.012]	.59 [.010]
SelSS	Trained SQE	.35 [.009]	.52 [.002]	.63 [.005]
	Deveaud *et al.* [4]	.40 [.003]	.66 [.001]	.77 [.005]
	ERisk-RF	**.41**$^{\triangle}$ [.002]	**.67**$^{\triangle}$ [.002]	**.79**$^{\triangle}$ [.010]
	Best conf.	.36	.52	.63
	Oracle	.50	.85	.94

The method we present here uses 20 search strategies that the system learned to choose according to query features. This risk sensitive selective search is effective to increase overall effectiveness. Significant effectiveness improvement is about 23% when compared to L2R documents and about 10% when compared to other selective approaches on 3 adhoc TREC collections. Selective query expansion used only two search strategies, which limits the system options. On the other hand, in another study, we used $20,000$ search strategies which limits its practical usability. We show that the E-risk approach we presented is more appropriate, both to provide enough options to the system to choose among and to keep it manageable in terms of maintenance. This paper is an extended abstract of Mothe and Ullah CIKM 2021 paper [1]. Moreover, this research is patented. [5]

[5]EP3771996A1: Information retrieval device and method using set of search configurations pre-selected using efficiency and risk functions. https://worldwide.espacenet.com/patent/search/family/067956648/publication/EP3771996A1?q=19305984.7 Josiane Mothe & Md Zia Ullah