The Search Duel: A Response to a Strong Ranker

Peter Izsak
Technion, Israel
peteriz@tx.technion.ac.il

Fiana Raiber
Technion, Israel
fiana@tx.technion.ac.il

Oren Kurland
Technion, Israel
kurland@ie.technion.ac.il

Moshe Tennenholtz
Microsoft Research and
Technion, Israel
moshet@ie.technion.ac.il

ABSTRACT

How can a search engine with a relatively weak relevance ranking function compete with a search engine that has a much stronger ranking function? This dual challenge, which to the best of our knowledge has not been addressed in previous work, entails an interesting bi-modal utility function for the weak search engine. That is, the goal is to produce in response to a query a document result list whose effectiveness does not fall much behind that of the strong search engine; and, which is quite different than that of the strong engine. We present a per-query algorithmic approach that leverages fundamental retrieval principles such as pseudo-feedback-based relevance modeling. We demonstrate the merits of our approach using TREC data.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

Keywords: search engine competition, dueling algorithms

1. INTRODUCTION

We revisit the classic ad hoc relevance ranking problem from a competition perspective. Rather than addressing a single ranking system, we consider a duel in which a search problem — i.e., a query representing an information need — is presented to two players. One of the players has a relevance ranking function that is considerably "weaker" (i.e., less effective) than that of the other player. Yet, his goal is to produce a ranking that is *competitive* with that created by the player with the stronger ranking function.

On the theory side, this type of interaction is modeled in the context of the recently introduced dueling algorithms [9]. Our goal is to introduce a pragmatic manifestation in the context of an adversarial retrieval setting (e.g., the Web).

The ranking functions employed by leading Web search engines are remarkably effective. However, there is still much room for improving retrieval effectiveness due to various reasons. Many of these are related to the adversarial nature of the retrieval setting (e.g., search engine optimization efforts). Furthermore, it is impossible for a search engine to index all possible documents in a large-scale and dynamically changing collection such as the Web. This reality provides some hope for a search engine with a relatively weak ranking function to compete with a search engine that has a much stronger ranking function.

A potential approach to addressing the duel challenge is to try to explicitly learn the ranking function of the strong search engine. However, this approach falls short in our competitive setting. That is, some of the most important information types utilized by the strong ranker may not be available to the weak ranker; e.g., those based on user engagement information such as clickthrough data.

We propose a per-query competitive approach. We let the weak search engine observe the output (i.e., the result list of the most highly ranked documents) of the strong search engine for a query. Using this list, which is treated as a pseudo feedback set, we induce a relevance model [10]. The model is used to modify the ranking of the weak search engine. The modification is based on a *bi-modal* criteria: retrieval effectiveness (in terms of relevance) and diversification with respect to the results presented by the strong search engine. The motivation for diversification is based on the following realization. Users of the strong search engine would have no incentive to switch to (or consider) the weak engine if they are presented with the same results.

Empirical evaluation performed with TREC data attests to the effectiveness of our approach. For example, we show that the approach can be used to boost the retrieval effectiveness of weak rankers to a level competitive with that of strong rankers, while maintaining relatively low overlap with the strong rankers' result lists. The approach also substantially outperforms a highly effective fusion method that merges the results of the strong and weak search engines.

This paper describes a preliminary, and the first (to the best of our knowledge), attempt to address the interesting and practical challenge of a search engine duel. Naturally, an abundance of research challenges, in addition to those we address here, arise. We discuss some of these in Section 5.

2. RELATED WORK

As mentioned, from a theory perspective, the work on dueling algorithms [9] deals with a setting similar to ours. Although the emphasis is on computing minimax strategies, the basic building block is computing the response of one

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.
Copyright 2014 ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609474.

agent to another. However, the model is stylized and does not refer to the realistic search duel we discuss here.

There is a large body of work on merging document lists that were retrieved in response to a query from the same corpus [4] or from different corpora [1]. Our use of a relevance model induced from one list to re-rank another list is conceptually reminiscent of work on using inter-document similarities between two lists for re-ranking [11]. However, in contrast to all these results merging approaches which aim to maximize only relevance, our methods are designed to also minimize overlap with the strong engine's result list.

There is work on training a ranker for one domain (language) and applying it to another [7, 8]. In contrast, we do not assume different domains or languages and we do not train a ranker but rather use a pseudo-feedback-based relevance model.

We note that existing methods for diversifying search results (e.g., [2, 12]) focus on a single retrieved list and on a notion of diversification — i.e., coverage of query aspects — which is different than that we address here; that is, the overlap with another result list.

3. SEARCH ENGINE RESPONSES FRAMEWORK

Let q be a query which is *fixed* here and after. Let $\mathcal{C}$ be a corpus of documents upon which two search engines perform a search in response to q. One of the search engines, henceforth referred to as *strong*, is assumed to have a more effective relevance ranking function than that of the other — the so called *weak* engine. Specifically, the ranking induced by the strong engine in response to q is assumed to be of higher effectiveness than that induced by the weak engine.

We use $L^{[n]}_{strong}$ and $L^{[n]}_{weak}$ to refer to the lists of the n documents that are the most highly ranked by the strong and weak engines, respectively. The goal we pursue is devising an *effective response strategy* for the weak engine, given that it has access to the list $L^{[n]}_{strong}$ of the strong engine.[1] By response we mean producing a new result list, $L^{[n]}_{weak;response}$, composed of n documents, that will replace the original list, $L^{[n]}_{weak}$; yet, $L^{[n]}_{weak}$ can be used to produce the response.

A key question is what makes a response "effective". Obviously, $L^{[n]}_{weak;response}$ should be of the highest possible effectiveness, preferably, not significantly lower than that of $L^{[n]}_{strong}$. Supposedly, then, a highly effective response is setting $L^{[n]}_{weak;response} \overset{def}{=} L^{[n]}_{strong}$; that is, having the weak engine replicate the result list produced by the strong engine. However, assuming that the strong search engine already has a well established, and wide, base of users, such a response is not likely to result in any incentive for users to switch engines. Therefore, the second criterion we set for an effective response is that the overlap, in terms of shared documents, between $L^{[n]}_{weak;response}$ and $L^{[n]}_{strong}$ will be minimal; i.e., the goal is to differentiate the weak engine from the strong engine.

In summary, the weak engine should produce a result list that is as diverse as possible, and competitive in terms of effectiveness, with respect to the result list produced by the strong engine. In doing so, the weak engine can use its original list $L^{[n]}_{weak}$ and that of the strong engine, $L^{[n]}_{strong}$.

3.1 Relevance Modeling as a Basis for Response Strategies

The basic assumption underlying the strategies that we employ below is that there are relevant documents in $L^{[n]}_{weak}$ that are not in $L^{[n]}_{strong}$. The reason could be, for example, a different coverage of the indexes of the two engines. Yet, it is not necessarily the case that these documents are ranked high enough due to the relatively weak relevance ranking function of the weak engine. Thus, we use a relevance language model R [10] induced from $L^{[k]}_{strong}$ — the k ($\leq n$) documents that are the highest ranked in $L^{[n]}_{strong}$ — to re-rank $L^{[n]}_{weak}$; $L^{[n]}_{weak;re-rank}$ denotes the resultant ranked list. As R reflects a model of relevance of the strong engine with respect to the query q, we assume that the ranking of $L^{[n]}_{weak;re-rank}$ is of higher effectiveness than that of $L^{[n]}_{weak}$. We provide empirical support to this assumption in Section 4. The re-ranking of $L^{[n]}_{weak}$ using R is based on the cross entropy between R and the language models induced from the documents. Details regarding (relevance) language model induction are provided in Section 4.1.

In what follows we present several strategies of producing the final result list of the weak engine, $L^{[n]}_{weak;response}$, using the lists $L^{[n]}_{weak;re-rank}$ and $L^{[n]}_{strong}$.

3.2 Response Strategies

The first response strategy, **WeakReRank**, is simply using $L^{[n]}_{weak;re-rank}$ for $L^{[n]}_{weak;response}$. The source for diversity with the strong list $L^{[n]}_{strong}$ is the assumed existence of documents in $L^{[n]}_{weak}$ that are not in $L^{[n]}_{strong}$. The presumably improved effectiveness of $L^{[n]}_{weak;re-rank}$ is due to the way it was created; that is, re-ranking $L^{[n]}_{weak}$ using a relevance model induced from $L^{[n]}_{strong}$.

The second response strategy is a probabilistic round robin procedure, henceforth referred to as **ProbRR**. We create $L^{[n]}_{weak;response}$ top down by scanning the lists $L^{[n]}_{weak;re-rank}$ and $L^{[n]}_{strong}$ also top down. The next document selected for $L^{[n]}_{weak;response}$ is taken from $L^{[n]}_{weak;re-rank}$ with probability p and from $L^{[n]}_{strong}$ with probability $1-p$. (Documents that were already selected are skipped.) Smaller values of p result in more documents taken from $L^{[n]}_{strong}$. Accordingly, the overlap of the final result list $L^{[n]}_{weak;response}$ with $L^{[n]}_{strong}$ can increase and the effectiveness is potentially maintained at the same level as that of $L^{[n]}_{strong}$. Hence, ProbRR enables a certain degree of control over the effectiveness and diversification of $L^{[n]}_{weak;re-rank}$ with respect to $L^{[n]}_{strong}$ using different values of p. However, $L^{[n]}_{weak;re-rank}$ can contain documents that are also in $L^{[n]}_{strong}$. Thus, higher values of p do not necessarily directly translate to increased diversity with respect to $L^{[n]}_{strong}$.

To directly control the level of diversity of $L^{[n]}_{weak;response}$ with respect to $L^{[n]}_{strong}$, we examine a variant of ProbRR termed **ProbResRR** — the third response strategy we propose. Instead of using $L^{[n]}_{weak;re-rank}$ and $L^{[n]}_{strong}$ we use

[1] In practical settings, the weak engine can potentially record, from time to time, the results produced by the strong search engine, specifically, in response to common queries.

$L^{[n]}_{weak;re-rank} \setminus L^{[n]}_{strong}$ and $L^{[n]}_{strong}$; i.e., we remove from $L^{[n]}_{weak;re-rank}$ documents that are in $L^{[n]}_{strong}$ and maintain the ranking of the residual documents. We then apply the same procedure described above for ProbRR to create the final result list $L^{[n]}_{weak;response}$. Using larger values of p results in increased diversity with respect to $L^{[n]}_{strong}$.

4. EVALUATION

4.1 Experimental Setup

We used the Web tracks of TREC 2009–2011, henceforth TREC-2009, TREC-2010, and TREC-2011, to create the strong vs. weak search engine setting. We focused on *runs* submitted for the ClueWeb09 category B collection which is composed of around 50 million documents.

We randomly selected 30 pairs of runs from all those submitted and which contain at least 1000 documents as results for each query in the track. In each pair of runs, the result lists of the run whose MAP (@1000) is higher serve for the lists of the strong engine, $L^{[n]}_{strong}$, and those of the run with the lower MAP serve for the lists of the weak engine, $L^{[n]}_{weak}$. Each result list contains $n = 1000$ documents. We report average performance over the 30 samples. Thus, here, the strong and weak engines are represented by "averages" over runs of which one is *on average* more effective than the other.

Titles of TREC topics serve for queries. Stopwords on the INQUERY list were removed from queries but not from documents. The Indri toolkit (www.lemurproject.org/indri) was used for experiments.

To evaluate retrieval performance, we use MAP (@1000) and NDCG (@20). Statistically significant differences of performance, computed over the 30 pairs of runs, were determined using the two-tailed paired t-test at a 95% confidence level. To measure the diversity of the result list of the weak engine with respect to that of the strong engine, we use the overlap (i.e., number of shared documents) at the top ten (OV@10) and twenty (OV@20) ranks of the two lists.

We use Dirichlet-smoothed document language models with the smoothing parameter set to 1000. For the relevance model R, we use the rank-based RM3 model [5] which is constructed from the top k (= 10) documents in the strong engine's result list. (We use ranks rather than retrieval scores as the latter are not assumed to be known.) The number of terms used by RM3, and its query anchoring parameter, are set to the default values of 50 and 0.5, respectively. Using this (under optimized) default parameter setting allows to demonstrate the *potential* of using the relevance modeling idea as a basis for producing responses.

As a reference comparison response strategy we use the highly effective **CombMNZ** fusion method [6] to merge the result lists of the weak and strong engines. Document scores induced from ranks, as suggested in [3], are used in CombMNZ. As all other fusion methods, CombMNZ addresses (explicitly) only one of the two criteria for effective response strategy — i.e., retrieval effectiveness.

4.2 Experimental Results

The performance numbers of all methods are presented in Table 1. We use ProbRR(p) and ProbResRR(p) to indicate that the ProbRR and ProbResRR response strategies were used with the probability parameter p.

Table 1 shows that Strong is much more effective than Weak for both MAP and NDCG; the differences are substantial and statistically significant for all experimental settings. Furthermore, the overlap between Strong and Weak, as measured by OV@10 and OV@20, is low. Thus, the experimental setting we used adheres to the problem definition: (i) the strong search engine is (much) stronger in terms of retrieval effectiveness, and (ii) the overlap between the result lists of the strong and weak search engines is not high.

We also see in Table 1 that WeakReRank is quite an effective response strategy; specifically, in comparison to a highly effective fusion method (CombMNZ) in terms of both retrieval effectiveness and overlap at top ranks with Strong. WeakReRank's retrieval effectiveness can be statistically significantly worse than that of Strong. However, WeakReRank outperforms Weak for MAP and NDCG, with all the improvements being statistically significant. Although the overlap at top ranks of WeakReRank with Strong is larger than that of Weak, it is still quite low with respect to that of the other strategies considered. These findings attest to the effectiveness of using relevance modeling based on the result list of the strong search engine so as to re-rank the result list of the weak search engine.

Table 1 also shows that ProbRR is a highly effective response strategy in many cases. Evidently, the balance between retrieval effectiveness and overlap with Strong can be effectively controlled via the parameter p. Although in terms of overlap with Strong, ProbRR is somewhat less effective than WeakReRank and CombMNZ, in terms of retrieval effectiveness it is substantially better than the two and often better than Strong.

It is also evident in Table 1 that ProbResRR is less effective than ProbRR for MAP and NDCG. However, the overlap numbers for ProbResRR are lower than those for ProbRR. This finding is not surprising because ProbRR uses the re-ranked list of the weak engine while ProbResRR uses the residual documents in the same list that remain after the removal of documents that also appear in the result list of the strong engine.

The effectiveness-diversity tradeoff. We next study the effect of the parameter p used in ProbRR and ProbResRR on the tradeoff between the retrieval effectiveness of the response list of the weak search engine, as measured using MAP, and its overlap with the result list produced by the strong search engine, measured using OV@10.

Figure 1 presents the results of setting p to values in $\{0, 0.1, \ldots, 1\}$; $p = 0$ amounts to using only the result list of the strong search engine in ProbRR and ProbResRR, while $p = 1$ amounts to using the result list $L^{[n]}_{weak;re-rank}$.

We can see that for ProbResRR, MAP decreases with increasing values of p. The reason is that fewer documents that appear in the result list of the strong engine are used. Furthermore, ProbRR outperforms ProbResRR for all $p > 0$. In addition, we see that ProbRR can attain its optimal performance for $0 < p < 1$ (i.e., outperform both Strong and WeakReRank) which echoes findings in work on fusion [4].

For both ProbRR and ProbResRR, OV@10 decreases with increasing values of p, as fewer documents are selected from the result list of the strong engine. As expected, for the same value of p (> 0), the OV@10 of ProbRR is higher than that of ProbResRR. Yet, for the same value of overlap, the MAP of ProbRR is higher than that of ProbResRR.

	TREC-2009				TREC-2010				TREC-2011			
	MAP	NDCG	OV@10	OV@20	MAP	NDCG	OV@10	OV@20	MAP	NDCG	OV@10	OV@20
Strong	17.1_w	26.5_w	–	–	19.9_w	$\mathbf{25.2}_w$	–	–	19.3_w	28.0_w	–	–
Weak	11.4^s	19.5^s	**17.4**	**19.4**	13.6^s	16.9^s	**19.7**	**23.3**	13.1^s	21.6^s	**18.6**	**20.7**
WeakReRank	16.7_w	$\mathbf{30.3}^s_w$	35.0	34.5	16.5^s_w	19.8^s_w	29.3	30.5	17.6^s_w	27.9_w	30.3	31.1
CombMNZ	13.4^s_w	21.6^s_w	40.9	43.6	16.4^s_w	18.9^s_w	38.3	42.1	17.2^s_w	24.0^s_w	45.1	47.5
ProbRR(0.2)	18.3^s_w	27.8^s_w	87.2	87.1	$\mathbf{20.6}^s_w$	24.1^s_w	85.2	85.0	21.7^s_w	28.9^s_w	86.0	86.3
ProbRR(0.5)	$\mathbf{18.7}^s_w$	29.2^s_w	67.5	67.6	20.2_w	22.6^s_w	63.3	64.4	$\mathbf{22.3}^s_w$	29.6^s_w	65.6	66.1
ProbRR(0.7)	18.6^s_w	29.7^s_w	55.4	54.8	19.4_w	21.5^s_w	49.7	50.6	22.0^s_w	$\mathbf{29.7}^s_w$	51.4	52.8
ProbResRR(0.2)	15.4^s_w	23.7^s_w	81.0	80.5	17.9^s_w	21.8^s_w	79.4	79.8	19.2_w	26.2^s_w	80.3	80.0
ProbResRR(0.5)	11.2^s	18.4^s	52.2	52.4	12.8^s	16.0^s	50.0	50.5	15.4^s_w	21.7^s	50.4	50.2
ProbResRR(0.7)	8.2^s_w	13.9^s_w	32.6	33.2	9.0^s_w	11.7^s_w	29.9	29.8	12.0^s	17.6^s_w	30.4	30.6

Table 1: Main result table. The numbers in the parentheses for the ProbRR and ProbResRR strategies are the value of the p parameter. The highest result in a column for MAP and NDCG, and the lowest for OV@10 and OV@20, is boldfaced. 's' and 'w' mark statistically significant differences, for MAP and NDCG, with Strong and Weak, respectively.

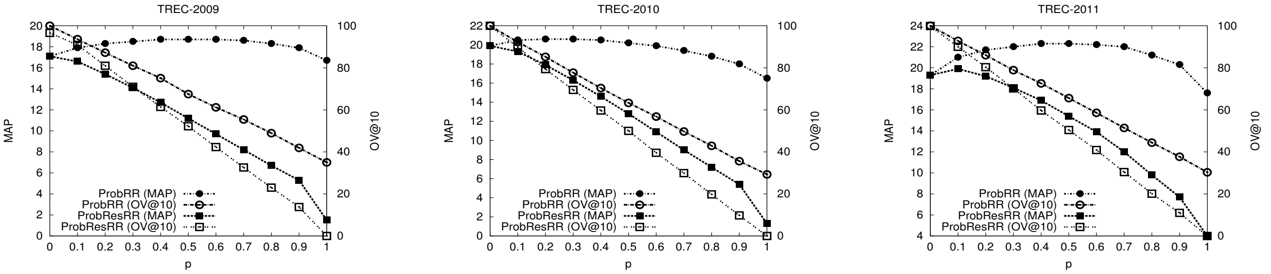

Figure 1: The effect of the parameter p on MAP and OV@10. The y-axis on the right of a figure is the range of OV@10. The y-axis on the left of a figure is the range of MAP.

Thus, we arrive to the conclusion that tuning the parameter p in ProbRR and ProbResRR helps to effectively control the balance between retrieval effectiveness and overlap (diversity) with the strong engine. Furthermore, ProbRR is a more effective response strategy than ProbResRR as it allows to attain improved level of retrieval effectiveness for the same level of overlap with the strong search engine.

5. CONCLUSIONS AND FUTURE WORK

We presented the first (preliminary) attempt to address the search engine duel problem; namely, how can a search engine with a relatively weak relevance ranking function compete with a search engine with a much stronger relevance ranking function? We devised an algorithmic response framework which consists of several strategies that can be used by the weak search engine. We consider a response to be effective if it results in improved search effectiveness and the search results are different than those presented by the strong engine. Empirical evaluation demonstrated the merits of our response strategies and shed some light on the (relevance) effectiveness-diversity tradeoff embodied in our bi-modal criteria for response effectiveness.

Devising additional criteria for response effectiveness, along with developing additional corresponding response strategies, is the first future venue we intend to explore. We also plan on devising response strategies for the strong engine.

Acknowledgments We thank the reviewers for their comments. This work has been supported by and carried out at the Technion-Microsoft Electronic Commerce Research Center. This work has also been supported in part by Microsoft Research through its Ph.D. Scholarship Program.

6. REFERENCES

[1] J. Callan. Distributed information retrieval. In W. Croft, editor, *Advances in information retrieval*, chapter 5, pages 127–150. Kluwer Academic Publishers, 2000.

[2] J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of SIGIR*, pages 335–336, 1998.

[3] G. V. Cormack, C. L. A. Clarke, and S. Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proc. of SIGIR*, pages 758–759, 2009.

[4] W. B. Croft. Combining approaches to information retrieval. In W. Croft, editor, *Advances in information retrieval*, chapter 1, pages 1–36. Kluwer Academic Publishers, 2000.

[5] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. A language modeling framework for selective query expansion. Technical Report IR-338, Center for Intelligent Information Retrieval, University of Massachusetts, 2004.

[6] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Proc. of TREC-2*, 1994.

[7] W. Gao, J. Blitzer, and M. Zhou. Using english information in non-english web search. In *Proc. of the 2nd ACM workshop on Improving Non English Web Searching, iNEWS*, pages 17–24, 2008.

[8] W. Gao, P. Cai, K.-F. Wong, and A. Zhou. Learning to rank only using training data from related domain. In *Proc. of SIGIR*, pages 162–169, 2010.

[9] N. Immorlica, A. T. Kalai, B. Lucier, A. Moitra, A. Postlewaite, and M. Tennenholtz. Dueling algorithms. In *Proc. of STOC*, pages 215–224, 2011.

[10] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proc. of SIGIR*, pages 120–127, 2001.

[11] L. Meister, O. Kurland, and I. G. Kalmanovich. Re-ranking search results using an additional retrieved list. *Information Retrieval*, 14(4):413–437, 2011.

[12] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proc. of WWW*, pages 881–890, 2010.