Is Learning to Rank Worth it? A Statistical Analysis of Learning to Rank Methods in the LETOR Benchmarks

Guilherme C. M. Gomes, Vitor C. Oliveira, Jussara M. Almeida, Marcos A. Gonçalves

Universidade Federal de Minas Gerais, Brazil
{gcm.gomes, vitorco, jussara, mgoncalv}@dcc.ufmg.br

Abstract. The Learning to Rank (L2R) research field has experienced a fast paced growth over the last few years, with a wide variety of benchmark datasets and baselines available for experimentation. We here investigate the main assumption behind this field, which is that the use of sophisticated L2R algorithms and models produce significant gains over more traditional and simple information retrieval approaches. Our experimental results in the LETOR benchmarks surprisingly indicate that many L2R algorithms, when put up against the best individual features of each dataset, may not produce statistically significant differences, even if the absolute gains may seem large. We also find that most of the reported baselines are statistically tied, with no clear winner.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: Learning to Rank

Keywords: Information Retrieval, Learning to Rank, Statistical Analysis

1. INTRODUCTION

Over the last few years, Learning to Rank (L2R) has become a very popular research topic, based on the general and well-accepted assumption that it produces a much better performance than traditional ranking methods, such as BM25 [Jones et al. 2000] or Language Based Models [Ponte and Croft 1998], in information retrieval tasks. Indeed, several new L2R methods [Liu 2011] and benchmark datasets, including large ones such as the LETOR [1] repository, have been developed and made available to the community, in recent years.

However, the development and efficient employment of such methods are not free of costs. Being based on supervised learning, they require labeled datasets in order to properly learn the ranking functions. Moreover, these datasets should be large and heterogeneous enough to be capable of representing the domains upon which they will be applied. Due to such strict requirements, constructing such datasets is not a trivial task. In fact, it is very costly. After building the required data, an usually very computationally demanding learning phase has to be applied to learn the ranking functions, which may also require an expensive parameter tuning for optimal performance. Finally, the use of such functions, in production mode in real search engines for example, is usually a two-stage process, in which traditional methods are first applied and, in a subsequent step, the more expensive learned function is used to re-rank the top results generated by the first step [Cambazoglu et al. 2010]. This implies in an additional overhead to produce query answers.

Given all these issues, as well as the continuous advance and interest in the area, we here take a

[1] http://research.microsoft.com/en-us/um/beijing/projects/letor/

This work is supported by the INWeb (CNPq 57.3871/2008-6) and by the authors' grants from CNPq and FAPEMIG.
Copyright©2013 Permission to copy without fee all or part of the material printed in JIDM is granted provided that the copies are not made or distributed for commercial advantage, and that notice is given that copying is by permission of the Sociedade Brasileira de Computação.

step back and reevaluate the main assumption upon which Learning to Rank built its foundations, which is that the use of sophisticated L2R algorithms and models produce significant gains over more traditional and simple information retrieval approaches. We also investigate whether there is one (or more) algorithm, out of various L2R techniques that have been proposed in the literature, that deliver superior effectiveness in most situations (e.g., different collections, different tasks, etc). In order to do so, we analyze the results of 13 methods, here referred to as baselines, over 6 large datasets of the LETOR 3.0 benchmark, as well as 6 baselines over 2 even larger datasets of the LETOR 4.0 benchmark, when put up against simple isolated feature rankers, using statistically significance tests. All the datasets and baseline results (but one) are available at the benchmark's web page. The only new method we tested that is not in the LETOR benchmark is a new implementation of a Random Forest ranker, which, as we shall see, produced some good results in some of our experiments. Our goal is to verify whether the effectiveness of these methods is better than that produced with the best feature of each dataset when used in isolation, as given by some measure of ranking quality (e.g., Mean Average Precision). We also contrast the performance of each method against each other using the same statistical methodology and datasets. To our knowledge no previous work has performed such detailed comparison with a rigorous statistical analysis.

Despite a work in progress, our experimental results already reveal that: (1) in most datasets, the best single feature, ranked by Mean Average Precision (MAP) or Normalized Discounted Cumulative Gain cut at the top 10 element (NDCG@10) [Järvelin and Kekäläinen 2002], produces results that are statistically tied to most of the reported baselines; (2) the absolute differences in effectiveness provided by the L2R algorithms, when compared to single feature rankers, may be large, but, in most cases, are not statistically significant; and (3) almost all the baselines have very similar performances, making it unlikely that there is an overall best L2R method. Therefore a clear advantage of L2R solutions may not be confirmed in all situations, mainly considering the costs involved.

In comparison with its preliminary version [Gomes et al. 2012], this article brings new analyses of all benchmark algorithms in all datasets using a different evaluation metric, namely Normalized Discounted Cumulative Gain (NDCG) [Järvelin and Kekäläinen 2002]. NDCG has interesting properties when compared to MAP such as an exponential decay of the cumulative gain of the metric in rank positions with relevant documents, a comparison with an "ideal ranking", among others. As such, conclusions reported in [Gomes et al. 2012], based on MAP results, do not necessarily hold for NDCG, motivating the use of this metric. We also include in our analysis a new, very recently proposed L2R algorithm that is not present in the LETOR benchmark, namely Random Forests. This new algorithm, as far as we know, has never been tested in the LETOR datasets before. However, it produces some of the best results in several of the LETOR collections (although still tied with some of the best single features in some cases). This is a new, original contribution of this article.

The remainder of this article is organized as follows: Section 2 describes work related to this article; Section 3 details our experiments and analyses; Section 4 reports our results; Section 5 concludes the article and describes our future work.

2. RELATED WORK

Despite the great interest in Learning to Rank in recent years, most of the related work focuses on proposing new algorithms for ranking or novel applications of existing ones. After the publication of the LETOR dataset [Liu et al. 2007], very few studies were made concerning the effects of these public datasets on the task of learning to rank.

In [Minka and Robertson 2008], the authors observed that the ways in which documents were selected for each topic of the LETOR benchmark presented on [Liu et al. 2007] show that the selection has (for each of the three corpora) a particular bias or skewness. This observation has some unexpected effects that may considerably influence any learning-to-rank exercise conducted on these datasets.

However, most of these problems were explained and corrected by the benchmark's authors in [Liu et al. 2008]. In [Liu 2011], a comparison of 7 learning to rank algorithms is made on the LETOR 3.0 benchmark. Each algorithm is compared with each other in terms of Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). In comparison with this previous study, we here compare 13 learning to rank algorithms against not only each other but also against using the best single feature in various datasets of the LETOR 3.0 and 4.0 benchmarks. Moreover, unlike most previous work, we use statistical tests to support our analyses and conclusions.

An interesting study is reported in [Silva et al. 2011] whose goal is not to improve L2R algorithms but instead to reduce the cost of producing labeled datasets for the learning process. An unexpected finding was that the reduced training sets produced better results than using the whole datasets with some of the tested rankers in the LETOR datasets. This result points towards an interesting line of research on removing noise and redundancy in L2R datasets.

However, none of these previous efforts effectively evaluated the real gains of learning algorithms over traditional methods, like BM25 or Language Models, expressed as features of the dataset. Moreover, to our knowledge, the first effort towards a statistical comparison of learning to rank algorithms and an evaluation of their differences against the best single features was our previous work [Gomes et al. 2012]. This work extends this preliminary analysis by performing a more thorough evaluation, considering different evaluation metrics and including one more recently proposed L2R algorithm, and thus reaching more solid conclusions.

3. EXPERIMENTAL SETUP

For our experimentations, we employed 8 datasets from the LETOR benchmark. Namely, we used the HP2003, HP2004, NP2003, NP2004, TD2003, and TD2004 datasets, from LETOR 3.0, based on the Gov web page collection. The first four collections are more related to navigational tasks, in which a single unique page is the sole best answer for a query, while the latter two are related to more traditional informational queries. We also use the larger MQ2007, MQ2008 datasets from LETOR 4.0, based on the Gov2 collection, which are also related to informational tasks. These datasets, which are also informational, are different from the previous LETOR 3.0 datasets, in the sense that they have thousands of queries with much fewer relevant documents per query. In contrast, the LETOR 3.0 informational datasets have fewer queries and a higher number of relevant documents per query. All datasets are divided into 5 folds, with the goal of performing a 5-fold cross-validation, that is, 3 folds are used for training, one (validation) for parameter tuning, and the remaining one for testing. In the following, we start by describing in more detail each group of datasets in Section 3.1 and then further describe our experimental methodology in Section 3.2.

3.1 Collections and Baselines

Each of the many datasets encapsulated by the LETOR 3.0 benchmark is composed of feature vectors for query-document pairs, along with a corresponding relevance judgment indicating whether the document is relevant or not for the query. There are 64 features per pair, which correspond to various pieces of information commonly used by traditional approaches (such as PageRank) or the result of directly applying simpler methods, such as TF*IDF, BM25 and language models, for estimating the document's relevance to the query. Considering all 6 datasets contained within this version of the benchmark, we find 575 queries and over 580.000 labeled documents.

Also available on the benchmark's web page, we find 12 different baselines, namely: FRank, ListNet, AdaRank-MAP, AdaRank-NDCG, RankBoost, RankSVM, RankSVM-Struct, RankSVM-Primal, Regression, Regression+L2reg, Smooth Rank and SVMMAP. Aside from FRank and RankBoost, all algorithms use linear ranking functions.

Similar to the previous benchmark, LETOR 4.0 uses the same structure for its datasets, but with 46 features, instead of 64, per vector. The largest TREC datasets available, from the Million Query Tracks of 2007 and 2008, named as MQ2007 and MQ2008, are based on the Gov2 webpage collection. Both datasets sum over 2.500 queries and roughly 420.000 documents. Unlike in LETOR 3.0, we here find only 5 reported baseline algorithms: AdaRank-MAP, AdaRank-NDCG, ListNet, RankBoost and RankSVM-Struct. Any other information pertaining these baselines and datasets, as well as the datasets themselves, are available at the LETOR website.

Aside from the reported LETOR benchmark algorithms, we have also performed experiments of our own using Random Forests, as implemented in the RankLib package [2], and included its results in our analyses, making it the 13th and 6th baseline method for LETOR 3.0 and 4.0, respectively. As we shall see, its performance is very similar to the other methods, even though the Random Forest approach is quite different from the other baselines, making it an interesting comparison candidate. As for the parameterization, we used only the validation sets, as done for the LETOR baselines.

3.2 Choice of the Best Feature

For comparison purposes, we performed the following feature selection procedure. For each fold, the values of each feature of each query-document pairs of the test set are extracted and used as a ranking score for its respective pair, thus obtaining a number of ranked lists equal to the number of features used to describe each document. Afterwards, we use the evaluation tool provided by the benchmark to calculate traditional information retrieval metrics, such as Mean Average Precision (MAP) and average Normalizing Discounted Cumulative Gain (NDCG) [Järvelin and Kekäläinen 2002], of the previously produced ranked lists. We then select the feature responsible for the best ranked list, in terms of generated MAP and average NDCG@10, to use as our isolated ranking feature.

We here compare both MAP and average NDCG@10 results with those obtained by the L2R baselines in the same test sets, using statistical significance tests with a 95% confidence level, aiming at quantifying the differences and verifying whether they are statistically significant. Specifically, to support our analyses and conclusions, we perform a pairwise comparison of all methods, applying paired difference tests [Jain 1991] for each pair, to verify whether they are statistically different.

4. RESULTS

We start by showing the best results obtained with a single feature for each considered dataset and comparing these results with the analyzed L2R algorithms. In order to perform a more thorough investigation on the effectiveness of the Learning to Rank algorithms, we first evaluate results of the single best feature ranking method when compared to the L2R baselines using MAP (Section 4.1) and NDCG@10 (Section 4.2) as our feature selection metric. Next, we compare all baselines using both MAP (Section 4.3) and NDCG@10 (Section 4.4).

4.1 Feature Ranking Results - MAP

As a result of our feature selection procedure, a single attribute was selected as the best one in each dataset. Some features were selected for more than one dataset. Table I shows the MAP scores generated by the evaluation tools for each single feature, as well as the feature name and its position in the document vectors. In particular, it is interesting to notice the lower MAP values in the TD collections, which are known to be difficult informational datasets.

Table II shows the relative MAP difference between the best ranking feature of each dataset and the L2R baselines reported for LETOR 3.0 as well as the Random Forest ranker. Next to the dataset's

[2]http://people.cs.umass.edu/ vdang/ranklib.html

Table I. Average MAP scores for the best ranked features of the different datasets, LETOR 3.0

Dataset	MAP Score	Best Feature
HP2003	0.7031	Hyperlink base feature propagation: weighted in-link (46)
HP2004	0.6171	Hyperlink base feature propagation: weighted in-link (46)
NP2003	0.5784	IDF of the URL (9)
NP2004	0.5202	Sitemap based score propagation (42)
TD2003	0.1973	Hyperlink base feature propagation: weighted in-link (46)
TD2004	0.1844	Sitemap based score propagation (42)
MQ2007	0.4534	LMIR.DIR of whole document (39)
MQ2008	0.4712	LMIR.DIR of whole document (39)

Table II. Relative MAP comparison between feature ranking and L2R algorithms, LETOR 3.0

Dataset	**HP2003 (46)**	**HP2004 (46)**	**NP2003 (9)**	**NP2004 (42)**	**TD2003 (42)**	**TD2004 (42)**
AdaRank-MAP	9.65% (+)	16.97% (+)	17.28% (+)	19.56% (=)	15.70% (=)	18.72% (=)
AdaRank-NDCG	6.38% (=)	12.02% (=)	15.46% (+)	20.51% (=)	20.00% (=)	5.00% (=)
FRank	0.90% (=)	10.45% (+)	14.8% (+)	15.49% (=)	2.93% (=)	29.56% (=)
ListNet	8.9% (=)	11.78% (=)	19.22% (+)	29.17% (+)	39.52% (+)	21.04% (+)
RankBoost	4.25% (+)	1.29% (=)	22.31% (+)	8.42% (=)	15.23% (=)	41.77% (+)
RankSVM	5.35% (+)	8.15% (=)	20.28% (+)	26.64% (+)	33.17% (=)	21.36% (=)
RankSVM-Primal	8.72% (+)	8.75% (=)	19.00% (+)	29.85% (+)	34.44% (=)	11.81% (=)
RankSVM-Struct	8.45% (+)	9.93% (=)	17.36% (+)	30.17% (+)	37.48% (=)	19.1% (+)
Regression	-29.35% (-)	-14.84% (-)	-2.42% (=)	-1.16% (=)	22.08% (=)	12.72% (=)
Regression-L2reg	6.47% (=)	2.09% (=)	17.98% (+)	31.98% (+)	23.33% (+)	8.023% (=)
SmoothRank	5.54% (=)	16.28% (=)	18.76% (+)	27.26% (+)	23.93% (+)	11.14% (=)
SVMMAP	8.56% (=)	16.24% (+)	20.31% (+)	29.95% (+)	36.61% (+)	26.14% (+)
Random Forest	9.41% (+)	2.18% (=)	21.39% (+)	15.32% (=)	38.35% (+)	38.3% (+)

name, in parenthesis, we find the best feature's identifier. When the performance of the L2R algorithm is statistically better than the single feature ranking, a (+) sign is provided after the MAP difference. When it is statistically equal, (=) is included, whereas a (-) sign indicates that the L2R method is statistically inferior to the single feature. In other words, a positive sign indicates that the L2R approach has a MAP score significantly higher, with 95% confidence, than the single feature, while a negative sign indicates a statistically significant lower score.

By looking at Table II, we see that, aside from the NP datasets, only half of the L2R algorithms are statistically superior to the isolated feature ranking in each dataset. In other words, in several cases the absolute differences in performance may not be statistically sound. In fact, despite some large *average* gains, there are a lot of statistical ties and even performance losses of the L2R algorithms, in comparison with the isolated feature rankings. For instance, in the HP2004 dataset, a difference (on average) of 16.28% of the SmoothRank algorithm over the best single feature is indeed not significant: both methods are tied with 95% confidence. This is very surprising as we expected that all or at least most of the algorithms would be able to effectively combine the features to deliver a better performance. However, the variability of the results is so large that relying only on average MAP to determine the best method is not enough. In contrast, a 10.45% gain of the FRank method over the best single feature in the same dataset is significant. Since the number of replications and confidence level are fixed (i.e., 5 and 95%, respectively), the lower variability inherent to FRank results allows us to conclude towards a significant difference.

Table III. Relative MAP comparison between feature ranking and L2R algorithms, LETOR 4.0

Dataset	MQ2007 (39)	MQ2008 (39)
AdaRank-MAP	0.96% (+)	1.11% (=)
AdaRank-NDCG	1.51% (+)	2.38% (=)
ListNet	2.61% (+)	1.34% (=)
RankBoost	2.83% (+)	1.34% (=)
RankSVM-Struct	2.45% (+)	-0.34% (=)
Random Forest	1.07% (=)	0.47% (=)

For the NP2003 and NP2004 datasets, most L2R algorithms (92.3% and 53.8% of the considered methods, respectively) are indeed statistically superior to the best single feature. However, it is interesting to note that, even in these datasets, some (apparently) large relative differences (e.g., 19.56%) are in fact not statistically significant with 95% confidence. It is also worth mentioning the large and significant losses (up to 29.35%) of the Regression method over the best feature in the two HP datasets. This may me due to the high correlations among several features in this dataset, which may be detrimental to this particular method as it relies on linear regression [Jain 1991].

Regarding the Random Forest algorithm, Table II shows that it outperforms the best single feature approach, with statistically significant gains, in 4 out of the 6 analyzed datasets, being statistically tied in the other two datasets. Some of the gains are quite large (over 38%), such as the ones in the TD2003 and TD2004 datasets.

Analogous to Table II, Table III presents results relative to LETOR 4.0. Similarly to the results found in LETOR 3.0, we here see cases where not only the relative differences are not statistically significant (with 95% confidence), such as in MQ2008, but also some cases where the gains are only marginal (e.g., 0.96% for AdaRank-MAP in the MQ2007 dataset). In fact, it is very surprising that in MQ2008, no method is able to outperform the best feature in isolation.

We note that, for both LETOR 3.0 and 4.0, the Random Forest ranker tends to have a similar behavior as the best baseline, in terms of the gains over the best feature approach, in most datasets but HP2004, MQ2007, and, to a lesser extent, NP2004. In other words, in all other datasets, the relative difference between the Random Forest algorithm and the best feature ranker tends to be (close to) the largest across all considered algorithms, even though no fine parameterization was performed, meaning that this algorithm has a lot of potential.

These results lead to interesting conclusions pertaining the effective gains of L2R and its aggregated costs. While the performed process of choosing the best features is not cost free, it is much cheaper than the complex machine learning algorithms. Indeed, we may not need to investigate all possible features. A smaller set of candidates could be used based on results reported in the literature.

4.2 Feature Ranking Results - NDCG@10

Like Table I, Table IV shows average NDCG@10 scores of the selected best feature for each dataset. Interestingly, 6 out of 8 selections match the ones made using MAP as our attribute selection metric. Moreover, 4 out of the 6 datasets in LETOR 3.0 have feature 46 (Hyperlink base feature propagation: weighted in-link) as the top ranking feature. Features that were chosen using both MAP and NDCG@10 are shown in bold in Table IV.

Table V shows the relative difference in average NDCG@10 and statistical significance (i.e., (+), (-) or (=)) between each L2R method and the single feature procedure in each dataset of LETOR 3.0. All values and statistical results were computed as in Table II. Algorithms that showed a similar significance both in MAP and NDCG@10 results are displayed in bold.

Note that 63 out of the 78 average NDCG@10 results (80.8%) reported in Table V have the same statistical behavior as the ones reported for MAP. Indeed, the same overall conclusions regarding the

Table IV. Average NDCG@10 scores for the best ranked features of the different datasets, LETOR 4.0

Dataset	NDCG@10 Score	Best Feature
HP2003	0.7910	**Hyperlink base feature propagation: weighted in-link (46)**
HP2004	0.7511	**Hyperlink base feature propagation: weighted in-link (46)**
NP2003	0.6907	LMIR.ABS of whole document (30)
NP2004	0.6125	**Sitemap based score propagation (42)**
TD2003	0.2637	**Hyperlink base feature propagation: weighted in-link (46)**
TD2004	0.2748	Hyperlink base feature propagation: weighted in-link (46)
MQ2007	0.4221	**LMIR.DIR of whole document (39)**
MQ2008	0.2230	**LMIR.DIR of whole document (39)**

Table V. Relative average NDCG@10 comparison between feature ranking and L2R algorithms, LETOR 3.0

Dataset	**HP2003 (46)**	**HP2004 (46)**	**NP2003 (30)**	**NP2004 (42)**	**TD2003 (46)**	**TD2004 (46)**
AdaRank-MAP	**6.00% (+)**	**10.88% (+)**	**10.63% (+)**	**22.40% (=)**	**16.37% (=)**	**19.54% (=)**
AdaRank-NDCG	**1.89% (=)**	**7.27% (=)**	**11.09% (+)**	20.55% (+)	**15.13% (=)**	**15.10% (=)**
FRank	0.75% (+)	1.39% (=)	**12.41% (+)**	**19.12% (=)**	**2.01% (=)**	**21.25% (=)**
ListNet	5.84% (+)	**4.45% (=)**	**16.09% (+)**	**32.70% (+)**	**32.10% (+)**	15.56% (=)
RankBoost	3.30% (=)	**-1.10% (=)**	**16.82% (+)**	**12.88% (=)**	**18.39% (=)**	**27.53% (+)**
RankSVM	2.12% (=)	**2.35% (=)**	**15.88% (+)**	**31.62% (+)**	31.25% (+)	**12.03% (=)**
RankSVM-Primal	**3.41% (+)**	**2.78% (=)**	**14.30% (+)**	**29.80% (+)**	35.42% (+)	**6.01% (=)**
RankSVM-Struct	3.18% (=)	**2.07% (=)**	**15.18% (+)**	**30.23% (+)**	31.47% (+)	12.47% (=)
Regression	**-24.86% (-)**	**-13.88% (-)**	**-3.58% (=)**	**6.70% (=)**	**23.74% (=)**	**10.3% (=)**
Regression-L2reg	**3.86% (=)**	**-4.30% (=)**	**16.19% (+)**	**31.27% (+)**	25.03% (=)	**3.06% (=)**
SmoothRank	**1.06% (=)**	**7.34% (=)**	**15.46% (+)**	**31.89% (+)**	**24.47% (+)**	**5.78% (=)**
SVMMAP	5.24% (+)	9.45% (=)	**15.63% (+)**	**31.83% (+)**	27.67% (=)	**21.67% (+)**
Random Forest	**4.98% (+)**	**-4.65% (=)**	**15.04% (+)**	**15.63% (=)**	**34.14% (+)**	**27.29% (+)**

Table VI. Relative NDCG@10 comparison between feature ranking and L2R algorithms, LETOR 4.0

Dataset	**MQ2007 (39)**	**MQ2008 (39)**
AdaRank-MAP	**2.71% (+)**	**2.45% (=)**
AdaRank-NDCG	**3.52% (+)**	**3.30% (=)**
ListNet	**5.20% (+)**	**3.12% (=)**
RankBoost	**5.77% (+)**	**0.97% (=)**
RankSVM-Struct	**5.17% (+)**	**2.05% (=)**
Random Forest	4.16% (+)	**2.00% (=)**

lack of a direct correspondence between large absolute gains and statistically significant differences, drawn based on MAP results, also hold for NDCG@10 results. For example, a relative difference of 3.86% of Regression-L2Reg over the best single feature ranking on HP2003 is indeed a statistical tie, whereas the marginal 0.75% average gains of FRank over the best feature approach on the same dataset - a much smaller relative difference - is a statistical win. Once again, the diverse variability inherent to each algorithm plays a key role in these conclusions. Overall, we find that the L2R algorithm and the best single feature ranking are statistically tied in 47.4% of the cases, whereas in only 51% of the cases the former is statistically superior to the much simpler and cheaper feature selection and ranking approach.

Table VI shows NDCG@10 results for the LETOR 4.0 benchmark. Once again, these results lead to very similar conclusions as the MAP results, reported in Table III, with the exception of Random Forest in MQ2007. Unlike observed for MAP, Random Forest is statistically superior to the best single

Table VII. Baselines' average MAP and confidence intervals for the different datasets, LETOR 3.0

Dataset	HP2003	HP2004	NP2003	NP2004	TD2003	TD2004
AdaRank-MAP	**0.771 ± 0.071**	**0.722 ± 0.103**	**0.678 ± 0.087**	**0.622 ± 0.055**	**0.228 ± 0.106**	0.219 ± 0.042
AdaRank-NDCG	**0.748 ± 0.125**	**0.691 ± 0.053**	**0.668 ± 0.103**	**0.627 ± 0.046**	**0.237 ± 0.129**	0.194 ± 0.035
FRank	0.709 ± 0.077	**0.682 ± 0.112**	**0.664 ± 0.082**	**0.601 ± 0.112**	**0.203 ± 0.089**	**0.239 ± 0.042**
ListNet	**0.766 ± 0.095**	**0.69 ± 0.104**	**0.690 ± 0.083**	**0.672 ± 0.094**	**0.275 ± 0.100**	0.223 ± 0.006
RankBoost	0.733 ± 0.089	0.625 ± 0.015	**0.707 ± 0.040**	0.564 ± 0.036	**0.227 ± 0.087**	**0.261 ± 0.034**
RankSVM	**0.741 ± 0.069**	**0.667 ± 0.099**	**0.696 ± 0.068**	**0.659 ± 0.108**	**0.263 ± 0.111**	0.224 ± 0.035
RankSVM-Primal	**0.764 ± 0.087**	**0.671 ± 0.096**	**0.688 ± 0.078**	**0.675 ± 0.122**	**0.265 ± 0.109**	0.206 ± 0.027
RankSVM-Struct	**0.763 ± 0.094**	**0.678 ± 0.084**	**0.679 ± 0.073**	**0.677 ± 0.090**	**0.271 ± 0.115**	0.220 ± 0.025
Regression	0.497 ± 0.042	0.526 ± 0.075	0.564 ± 0.095	0.514 ± 0.064	**0.241 ± 0.083**	0.208 ± 0.034
Regression-L2reg	**0.749 ± 0.101**	**0.63 ± 0.085**	**0.682 ± 0.068**	**0.687 ± 0.108**	**0.243 ± 0.100**	0.199 ± 0.024
Smooth Rank	**0.742 ± 0.109**	**0.718 ± 0.102**	**0.687 ± 0.064**	**0.662 ± 0.097**	**0.245 ± 0.084**	0.205 ± 0.024
SVMMAP	**0.763 ± 0.096**	**0.717 ± 0.077**	**0.696 ± 0.060**	**0.676 ± 0.071**	**0.270 ± 0.093**	0.233 ± 0.032
Random Forest	**0.769 ± 0.059**	0.631 ± 0.123	**0.702 ± 0.042**	**0.600 ± 0.072**	**0.273 ± 0.101**	**0.255 ± 0.045**

feature in terms of average NDCG@10 in that dataset.

4.3 Baseline Comparisons - MAP

We now turn to our second goal, which is to compare the supervised rankers in the used collections. Tables VII and VIII show average MAP results obtained for each baseline, jointly with the corresponding 95% confidence intervals. Best results for each dataset, along with statistical ties according to paired tests with 95% confidence, are shown in bold.

In the HP2003 dataset, there is a statistical tie for the best method among 10 out of 13 baselines, i.e., the differences among them are not statistically significant with 95% confidence. The worst method is Regression, which is inferior to all baselines and even to the best feature in isolation (difference to the best performer, AdaRankMap, of 35%). Notice however that the second worst method (FRank) is only at most 8% worse than the best performer. Thus, in general, except for Regression, the differences among all considered baselines are, if significant, relatively small. In the HP2004 dataset we also have statistically tied results for 10 baselines. The only baseline that is significantly inferior to the others is (once again) Regression. Unlike for HP2003, Random Forest did not perform well in this dataset. We here also observe some large differences that are not statistically significant, such as the gap between AdaRank-Map and Regression+L2reg (12.7%). As discussed before, these results clearly reflect the large variability of the methods across the various folds of the datasets.

Unlike in HP2004, all methods but Regression are statistically tied in the NP2003 dataset, making it once again impossible to single out a best ranking method. In the NP2004, we have a similar situation with 10 out of 13 methods statistically tied. Surprisingly, RankBoost, which has a good performance in the previous datasets and is one of two best rankers in TD2004 (see below), is tied with Regression as the worst method.

In the TD2003 dataset, all methods are tied, with no clear winner or loser. An interesting result found here is the very large relative differences (on average) of ListNet over FRank (26.2%), although they are still statistically tied with 95% confidence. In the last dataset of LETOR 3.0, TD2004, RankBoost, FRank, and Random Forest are tied as the best rankers, outperforming the other 10 baselines. In fact, this dataset is the only one with few (three) methods outperforming most of the other algorithms, with some large significant differences in some cases (up to 26%). In contrast, in the other datasets, almost all considered baselines are tied as the best rankers, with 95% confidence.

Turning our attention to the LETOR 4.0 benchmark, Table VIII shows that there are four statistically tied best ranker methods in the MQ2007 dataset, namely AdaRank-NDCG, ListNet, RankBoost

Table VIII. Relative MAP comparison between feature ranking and L2R algorithms, LETOR 4.0

Dataset	MQ2007	MQ2008
AdaRank-MAP	0.458 ± 0.021	**0.476 ± 0.052**
AdaRank-NDCG	**0.460 ± 0.024**	**0.482 ± 0.054**
ListNet	**0.465 ± 0.020**	**0.477 ± 0.053**
RankBoost	**0.466 ± 0.023**	**0.477 ± 0.050**
RankSVM-Struct	**0.464 ± 0.022**	0.470 ± 0.052
Random Forest	0.458 ± 0.017	**0.473 ± 0.038**

Table IX. Baselines' average NDCG@10 and confidence intervals for different datasets, LETOR 3.0

Dataset	HP2003	HP2004	NP2003	NP2004	TD2003	TD2004
AdaRank-MAP	**0.838 ± 0.065**	**0.833 ± 0.065**	**0.764 ± 0.073**	**0.750 ± 0.061**	**0.307 ± 0.129**	**0.328 ± 0.078**
AdaRank-NDCG	**0.806 ± 0.116**	**0.806 ± 0.047**	**0.767 ± 0.061**	0.738 ± 0.051	**0.304 ± 0.156**	**0.316 ± 0.071**
FRank	0.797 ± 0.063	0.762 ± 0.087	**0.776 ± 0.055**	**0.730 ± 0.106**	**0.269 ± 0.109**	**0.333 ± 0.065**
ListNet	**0.837 ± 0.073**	**0.784 ± 0.100**	**0.802 ± 0.065**	**0.813 ± 0.100**	**0.348 ± 0.112**	**0.318 ± 0.011**
RankBoost	**0.817 ± 0.084**	**0.743 ± 0.055**	**0.807 ± 0.036**	0.691 ± 0.101	**0.312 ± 0.120**	**0.350 ± 0.043**
RankSVM	0.808 ± 0.068	**0.769 ± 0.098**	**0.800 ± 0.073**	**0.806 ± 0.124**	**0.346 ± 0.133**	0.308 ± 0.024
RankSVM-Primal	**0.818 ± 0.082**	**0.772 ± 0.091**	**0.789 ± 0.059**	**0.795 ± 0.110**	**0.357 ± 0.127**	0.291 ± 0.033
RankSVM-Struct	**0.816 ± 0.094**	**0.767 ± 0.102**	**0.795 ± 0.044**	**0.798 ± 0.108**	**0.347 ± 0.139**	0.309 ± 0.028
Regression	0.594 ± 0.061	0.647 ± 0.129	0.666 ± 0.092	0.654 ± 0.149	**0.326 ± 0.107**	0.303 ± 0.037
Regression-L2reg	**0.822 ± 0.119**	0.719 ± 0.071	**0.802 ± 0.046**	**0.804 ± 0.106**	**0.330 ± 0.144**	0.283 ± 0.031
Smooth Rank	**0.799 ± 0.104**	**0.806 ± 0.098**	**0.797 ± 0.052**	**0.808 ± 0.137**	**0.328 ± 0.127**	0.291 ± 0.028
SVMMAP	**0.832 ± 0.072**	**0.822 ± 0.053**	**0.799 ± 0.046**	**0.807 ± 0.092**	**0.337 ± 0.113**	**0.334 ± 0.032**
Random Forest	**0.830 ± 0.077**	0.716 ± 0.095	**0.795 ± 0.040**	0.708 ± 0.135	**0.354 ± 0.098**	**0.350 ± 0.048**

and RankSVM-Struct, whereas AdaRank-MAP and Random Forest are clear losers. A similar scenario is found in the MQ2008 dataset, but this time RankSVM-Struct is the worst performer and the only one not statistically tied with the others.

In general, we find that, in the vast majority of the analyzed datasets, most of the baselines are statistically tied, with no clear winner, raising a question of whether it is cost-effective to invest on developing new learning-to-rank algorithms, as opposed to combining multiple methods into a single hybrid solution or investing on reducing the costs (particularly in cases where the L2R methods outperform the single best feature).

4.4 Baseline Comparisons - NDCG10

Tables IX and X show the comparison among the various considered baselines in LETOR 3.0 and LETOR 4.0, respectively, in terms of average NDCG@10. Once again, results in bold indicate the best rankers for each dataset. These results contribute to strenghthen the conclusions drawn in Section 4.3. In particular, we find that almost all algorithms are statistically tied as the best rankers (77%) in all datasets. Even in the TD2004 dataset, where the best rankers in terms of MAP are FRank, RankBoost and Random Forest, in terms of average NDCG@10, around half of the reported baselines are statistically tied as best solutions. Similarly, nearly all algorithms in TD2003 and NP2003, aside from RankSVM and Regression, respectively, are tied as best rankers, further raising the question of whether or not there exists an undisputed winner, both by absolute values and statistical significance.

Finally, it is worth noting that Random Forest has a very similar behavior to the one reported in the previous section. Indeed, considering all MAP and NDCG results, Random Forest is tied with the best ranker in 10 out of all 16 evaluated results, showing the potential of this method.

Table X. Baselines' average NDCG@10 and confidence intervals for different datasets, LETOR 4.0

Dataset	MQ2007	MQ2008
AdaRank-MAP	0.433 ± 0.028	**0.229 ± 0.055**
AdaRank-NDCG	0.437 ± 0.029	**0.231 ± 0.055**
ListNet	**0.444 ± 0.027**	**0.230 ± 0.056**
RankBoost	**0.446 ± 0.029**	**0.226 ± 0.053**
RankSVM-Struct	**0.444 ± 0.031**	**0.228 ± 0.054**
Random Forest	0.440 ± 0.024	**0.228 ± 0.048**

5. CONCLUSIONS AND FUTURE WORK

After almost a decade of research and development of L2R algorithms, we have here raised two controversial but important questions that should be further discussed by the Information Retrieval community. First, given all the costs involved in L2R (e.g., labeling, training, tuning) and the overhead introduced by applying such techniques, for instance, for re-ranking search top results at query time, the cost-benefit ratio of applying such algorithms should be further investigated. Secondly, given the similar performance, with 95% confidence, of most of the 13 selected L2R algorithms across various different datasets, researching and developing new L2R algorithms may not be worth the effort. Indeed, although in a few datasets there are undisputed best rankers, it is not the case in most analysed datasets, in neither metric considered here (i.e., MAP and average NDCG@10).

Rather than providing definitive answers, our goal here, as work in progress, is to instigate discussion and re-evaluation of many L2R algorithms after having applied solid statistical methods in our own investigations of the subject. As future work, we intend to expand this study to consider even larger datasets, such as the ones provided by Microsoft Learning to Rank and Yahoo! Labs; new L2R algorithms, ranking metrics as well as to consider the posed question in other domains in which L2R techniques can be applied.

REFERENCES

Cambazoglu, B., Zaragoza, H., Chapelle, O., Chen, J., Liao, C., Zheng, Z., and Degenhardt, J. Early Exit Optimizations for Additive Machine Learned Ranking Systems. In *Proceedings of the International Conference on Web Search and Data Mining*. New York, USA, pp. 411–420, 2010.

Gomes, G. C. M., de Oliveira, V. C., de Almeida, J. M., and Gonçalves, M. A. Is Learning to Rank Worth it? A Statistical Analysis of Learning to Rank Methods. In *Proceedings of the Brazilian Symposium on Databases*. SBC, São Paulo, Brasil, pp. 193–200, 2012.

Jain, R. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, New York, USA, 1991.

Järvelin, K. and Kekäläinen, J. Cumulated Gain-Based Evaluation of IR Techniques. In *ACM Transactions on Information Systems*. New York, USA, pp. 422–446, 2002.

Jones, K. S., Walker, S., and Robertson, S. E. A Probabilistic Model of Information Retrieval: development and comparative experiments. *Information Processing & Management* 36 (6): 779–840, 2000.

Liu, T., Xu, J., Qin, T., and Li, H. How to Make LETOR More Useful and Reliable. In *Proceedings of the SIGIR Workshop on Learning to Rank for Information Retrieval*. pp. 52–58, 2008.

Liu, T., Xu, J., Qin, T., Xiong, W., and Li, H. LETOR: benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of the SIGIR Workshop on Learning to Rank for Information Retrieval*. pp. 3–10, 2007.

Liu, T.-Y. *Learning to Rank for Information Retrieval*. Springer, New York, USA, 2011.

Minka, T. and Robertson, S. Selection Bias in the LETOR Datasets. In *Proceedings of the SIGIR Workshop on Learning to Rank for Information Retrieval*. pp. 48–51, 2008.

Ponte, J. M. and Croft, W. B. A Language Modeling Approach to Information Retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Melbourne, Australia, pp. 275–281, 1998.

Silva, R., Gonçalves, M. A., and Veloso, A. Rule-Based Active Sampling for Learning to Rank. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*. Athens, Greece, pp. 240–255, 2011.