Inf Retrieval (2013) 16:584–628
DOI 10.1007/s10791-012-9209-9

# The whens and hows of learning to rank for web search

Craig Macdonald · Rodrygo L. T. Santos · Iadh Ounis

Received: 6 June 2011 / Accepted: 2 August 2012 / Published online: 2 September 2012

**Abstract** Web search engines are increasingly deploying many features, combined using learning to rank techniques. However, various practical questions remain concerning the manner in which learning to rank should be deployed. For instance, a sample of documents with sufficient recall is used, such that re-ranking of the sample by the learned model brings the relevant documents to the top. However, the properties of the document sample such as *when* to stop ranking—i.e. its minimum effective size—remain unstudied. Similarly, effective listwise learning to rank techniques minimise a loss function corresponding to a standard information retrieval evaluation measure. However, the appropriate choice of *how* to calculate the loss function—i.e. the choice of the learning evaluation measure and the rank depth at which this measure should be calculated—are as yet unclear. In this paper, we address all of these issues by formulating various hypotheses and research questions, before performing exhaustive experiments using multiple learning to rank techniques and different types of information needs on the ClueWeb09 and LETOR corpora. Among many conclusions, we find, for instance, that the smallest effective sample for a given query set is dependent on the type of information need of the queries, the document representation used during sampling and the test evaluation measure. As the sample size is varied, the selected features markedly change—for instance, we find that the link analysis features are favoured for smaller document samples. Moreover, despite reflecting a more realistic user model, the recently proposed ERR measure is not as effective as the traditional NDCG as a learning loss function. Overall, our comprehensive experiments provide the first empirical derivation of best practices for learning to rank deployments.

**Keywords** Learning to rank · Evaluation · Web search · Sample size · Document representations · Loss function

C. Macdonald (✉) · R. L. T. Santos · I. Ounis
School of Computing Science, University of Glasgow, Scotland, UK
e-mail: craig.macdonald@glasgow.ac.uk

R. L. T. Santos
e-mail: rodrygo@dcs.gla.ac.uk

I. Ounis
e-mail: iadh.ounis@glasgow.ac.uk

## 1 Introduction

Learning to rank (Liu 2009) is gaining increasing attention in information retrieval (IR), with machine learning techniques being used to learn an appropriate combination of features into an effective ranking model. An increasing amount of research is devoted to developing efficient and effective learning techniques, while major search engines reportedly deploy ranking models consisting of hundreds of features (Pederson 2010; Segalovich 2010).

Nevertheless, the manner in which learning to rank is deployed within real settings has not seen published discussion. For instance, learning to rank involves the use of a *sample* of top-ranked documents for a given query (Liu 2009), which are then re-ranked by the learned model before display to the user. However, in the literature, the properties of an effective sample are not clear. Indeed, despite his thorough treatment of existing learning to rank techniques, Liu (2009) does not address in detail how the sample should be made within an existing deployment, what document representation should be deployed when generating the sample (e.g. in addition to the body of the document, should anchor text be included or not?), nor how many documents it should contain.

Typically, a standard weighting model, such as BM25 (Robertson et al. 1992), is used to rank enough documents to obtain sufficient recall. Yet there is a great variation in the literature and existing test collections about how many documents should be re-ranked when learning models or deploying previously learned models, with large sample sizes such as “tens of thousands” (Chapelle et al. 2011), 5,000 (Craswell et al. 2010), 1,000 (Qin et al. 2009) as well as small samples such as 200 (Zhang et al. 2009) or even 20 (Chapelle and Chang 2011) observed. However, as we will show in this paper, such small samples can result in learned models with significantly degraded effectiveness.

On the other hand, reducing the size of the sample has various efficiency benefits. In particular, if document-at-a-time (DAAT) matching techniques such as Wand (Broder et al. 2003) are used to identify the sample of $K$ documents within the search engine, then using a smaller $K$ (i.e. a smaller sample) can markedly increase efficiency compared to a larger $K$ (Broder et al. 2003). Moreover, the number of feature computations are decreased for smaller $K$. Lastly, for environments where learning time is critical, the use of smaller samples markedly reduces the learning time of many learning to rank techniques such as AFS (Metzler 2007) and RankBoost (Freund et al. 2003).

Another issue concerns the loss function that learning to rank techniques deploy to produce effective models. In particular, *listwise* techniques, which directly use an IR evaluation measure for the loss function are often the most effective (Liu 2009). However, the choice of this *learning evaluation measure* and the rank cutoff to which it is computed during learning may have a noticeable impact on the effectiveness of the learned model on unseen data. Nonetheless, there has not been much work on such issues—for instance, while Robertson (2008) provided theoretical speculations, Donmez et al. (2009) observed that matching the learning measure with the test measure results in the best test accuracy in presence of “enough” training data. This was contradicted by Yilmaz and Robertson (2010), who observed that “enough” training data may not always be present. Overall, the lack of agreement in the existing literature suggests the necessity of a thorough empirical study.

Our work is the first study into best practices in a real deployment of learning to rank. In particular, the contributions of this work are threefold:

1. We propose an experimental methodology for investigating the size of the document sample as well as the choice and the rank cutoff of the evaluation measure deployed during learning;
2. We thoroughly experiment with multiple learning to rank techniques across several query sets covering both diverse information needs and corpora, namely the existing LETOR v3.0 GOV learning to rank test collection and the TREC Web track ClueWeb09 corpus;
3. From these experiments, we derive empirically identified recommendations on the sample size and the learning evaluation measure. In particular, three *research themes* are addressed in this paper, namely: the properties of the sample; the role of the learning evaluation measure; and the interaction between the learning evaluation measure cutoff and the sample size.

Through exhaustive experimentation across these three research themes, we investigate how the effectiveness of the learned model is affected by:

- the document representation used to generate the sample and the size of the sample;
- the learning to rank technique and the sample size;
- the type of information need and the sample size;
- the learning evaluation measure;
- the rank cutoff of the learning evaluation measure.

Among many conclusions, we find, for instance, that on the larger ClueWeb09 corpus the minimum effective sample can be as low as 10–20 documents for the TREC 2009 and 2010 Web track queries. However, surprisingly, a sample size of 1,500 documents is necessary to ensure effective retrieval for navigational information needs on the same corpus. Moreover, for the same navigational information needs, we also show the importance of including anchor text within the document representation used to generate the sample. Finally, the test evaluation measure is shown to be important, as evaluation by the ERR cascade measure (Chapelle et al. 2009) (which penalises redundancy) is shown to permit smaller effective samples than for other measures such as NDCG (Järvelin and Kekäläinen 2002) and MAP (Buckley and Voorhees 2000). As for the choice of the learning loss function, despite reflecting a more realistic user model, ERR is not as effective as NDCG, even when ERR is the target test measure. Indeed, our results provide additional insights into the ongoing aforementioned debate on whether the test measure is actually the most suitable for learning.

The remainder of the paper is structured as follows: Sect. 2 discusses related work and elaborates on the research themes that we study in this article; Sect. 3 describes the proposed experimental methodology that allows hypotheses concerning our research themes for learning to rank techniques to be validated; In Sect. 4, we define the experimental setup; Experimental results and analysis follow in Sect. 5; Concluding remarks are made in Sect. 6.

## 2 Problem definitions

Learning to rank is the application of machine learning techniques to generate a *learned model* combining different document features in an information retrieval system (Liu 2009). For instance, learning to rank techniques are often applied by Web search engines, to combine various document weighting models and other query-independent features (Pederson 2008). The form of the model generated by different learning to rank techniques differs in nature: for some, it is a vector of weights for linearly combining each feature (Metzler 2007; Xu and Li 2007); for others it can represent a learned neural network (Burges et al. 2005), or a series of regression trees (Weinberger et al. 2010). Regardless of the applied technique, the general steps for obtaining a learned model using a learning to rank technique are the following (Liu 2009):

0. Pooling: For each query in a training set, documents for which human relevance assessments are obtained are identified using a *pooling* methodology. By combining a diverse set of retrieval systems for each query (Voorhees and Harman 2005), a high quality pool can be obtained.
1. Top K Retrieval: For a set of training queries, generate a *sample* of documents using an initial retrieval approach.
2. Feature Extraction: For each document in the sample, extract a vector of feature values. A feature is a binary or numerical indicator representing the quality of a document, or its relation to the query.
3. Learning: Learn a model by applying a learning to rank technique. Each technique deploys a different loss function to estimate the goodness of various combinations of features. Documents are labelled according to the relevance assessments identified in step (0).

In practice, step (0) may have been performed separately—for instance, by reusing relevance assessments created as part of the TREC evaluation forum—or can be integrated into the general learning process, using active learning approaches to select additional documents to be assessed (Donmez and Carbonell 2009; Long et al. 2010).

Once a learned model has been obtained from the above learning steps, it can be deployed within a search engine as follows:

4. Top K Retrieval: For an unseen test query, a sample of documents is generated in the same manner as in step (1),
5. Feature Extraction: As in step (2), a vector of feature values is extracted for each document in the sample. The set of features should be exactly the same as for step (2).
6. Learned Model Application: The final ranking of documents for the query is obtained by applying the learned model on every document in the sample, and sorting by descending predicted score.

Figure 1 illustrates steps (4)–(6) of deploying a learned model in a search engine setting.

Of all seven steps, feature extraction [steps (2) and (5)] defines the features deployed by the search engine. As there are countless possible features from the body of information retrieval (IR) literature, we assume a standard set of features, therefore the efficient and effective calculation of these features are not considered in this discussion. On the other hand, step (6) is a straightforward application of the learned model to the vectors of feature values for each document in the sample (e.g. calculating the dot product of the document's feature values with the feature weights of the learned model to obtain the final document score), and hence there is little that can be altered to vary efficiency or effectiveness. Therefore, in this paper, we concentrate on the generation of the sample [steps (1) and (4)] and on the actual learning process [step (3)]. In particular, the generation of a sample in step (1) is important, since for efficiency reasons, it is impractical to apply learning to rank on the entire corpus, nor even on the union of all documents containing any query term. Moreover, efficiency benefits may be obtained by minimising the size of the sample, both during learning and application.

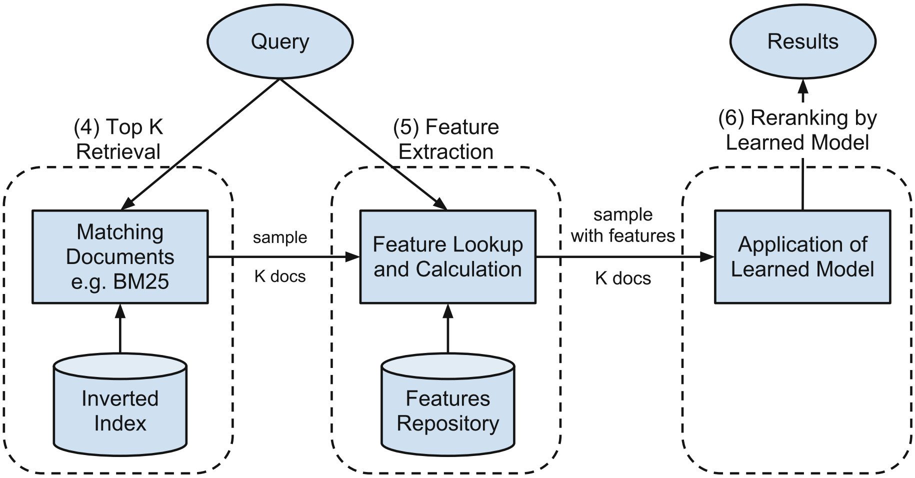


**Fig. 1** The three steps when deploying a learned model in a search engine setting

However, the appropriate properties of the sample have not been the subject of much investigation. Moreover, the effectiveness of the model generated in step (3) is of utmost importance. Indeed, learning to rank techniques that deploy standard IR evaluation measures as their loss functions are often among the most effective (Liu 2009). However, while the effectiveness of the learned model is reported to depend on the choice of the measure (Robertson 2008), this aspect has not seen much empirical treatment within the learning to rank literature, and—as mentioned in Sect. 1—even brought inconsistent observations (Donmez et al. 2009; Yilmaz and Robertson 2010). In the following, we expand on both the role of the sample (Sect. 2.1) and the role of the learning evaluation measure (Sect. 2.2), to form research themes around which hypotheses and research questions are postulated. The hypotheses and research questions of each research theme are later investigated in Sect. 5.

## 2.1 Sample

In this section, we define the motivations for sampling (Sect. 2.1.1), as well as reviewing the manner in which the sampling is performed in the literature (Sect. 2.1.2). We also review the tradeoff between the quality and the size of the sample (Sect. 2.1.3), as well as how the sample has been obtained in existing learning to rank test collections (Sect. 2.1.4). We use this discussion to motivate several hypotheses and research questions, which are defined in Sect. 2.1.4.

### *2.1.1 Why sample?*

The sample is a set of documents collected for each query, before learning [step (1)] or before applying a learned model [step (4)]. The motivations for the sample primarily occur for the efficient application of a learned model, but also has particular uses during learning, as detailed below. In the following, we provide the motivations for the use of sampling, ordered by their importance, across steps (4), (1) and also step (0).

*Sampling for Applying a Learned Model* [step (4)]: As mentioned above, a sample is used during the application of a learned model to reduce the size of the number of

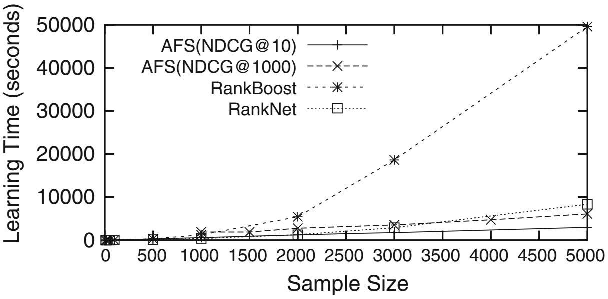


**Fig. 2** Learning time [c.f. step (3)] for several learning to rank techniques, namely Automatic Feature Selection (AFS) (Metzler 2007), RankBoost (Freund et al. 2003) and RankNet (Burges et al. 2005). Learning is applied for 75 features for a range of sample sizes

documents for which features are calculated (Liu 2009), by using an initial ranking approach to identify a set of documents that are likely to contain the relevant documents [Liu (2009) refers to the sample as a set of *"possibly relevant documents"*]. Minimising the number of documents for which features are calculated provides efficiency advantages, particularly if some features are expensive to compute [e.g. proximity features (Metzler and Croft 2005)]. Moreover, when a document-at-a-time (DAAT) retrieval strategy such as Wand (Broder et al. 2003) is used to identify the sample of documents in step (4), minimising the number of documents in the sample benefits efficiency by permitting Wand to omit the scoring of more documents that are unlikely to make the final sample (Broder et al. 2003).

*Sampling for Learning* [step (1)]: The use of a sample in learning has some similar motivations to its use within a deployed search engine. Indeed, even during learning, it is impractical to extract feature vectors for all documents in the corpus (Liu 2009). Following other supervised learning tasks such as classification and regression, a sample constitutes representative training data from which an effective learned model can be obtained. However, in contrast to the other supervised tasks, in learning to rank, it is not the case that every example document in the training data is independent and identically distributed (i.i.d). Indeed, the documents associated to each query in the sample form a group—the groups are i.i.d., however, the documents within a group are not i.i.d. (Li 2011). Instead, the documents should be identified in a deterministic manner (Li 2011). Finally, the learning time of many learning to rank techniques [step (3)] increases as the number of documents in the sample increases. Indeed, Fig. 2 shows the learning time for several learning to rank techniques as the sample size increases. For instance, for the AFS learning to rank technique (Metzler 2007) using NDCG@10 as the learning evaluation measure, the learning time for a sample of 2,000 documents is twice that for 1,000 documents (1,208 seconds vs. 604 seconds). Hence, if available resources for offline learning are limited, it may be desirable to reduce the learning time by constraining the size of the sample, providing there is no resulting loss in effectiveness when using a smaller sample.

*Sampling for Pooling* [step (0)]: Some work from the literature [e.g. Aslam et al. (2009)] merge the notion of sampling for learning [step (1)] with sampling for producing relevance assessments [step (0)]. In particular, the building of a high quality test collection involves the use of *multiple different* retrieval systems contributing documents to an assessment pool for each query. Each document in the pool is then assessed for its relevance. In this work, we assume that high quality relevance assessments are already

**Table 1** Example of biased sampling strategy

| Relevant | docid | BM25 | PageRank | URL-length |
|---|---|---|---|---|
| ✔ | 20 | 4.9 | 3.8 | 3 |
| × | 45 | 4.8 | 3.9 | 8 |
| ✔ | 19 | 4.3 | 3.7 | 2 |
| × | 8 | 4.1 | 3.3 | 5 |
| × | 38 | 4.0 | 3.3 | 3 |
| ... | | | | |
| ✔ | 7 | 0 | 1.3 | 5 |
| ✔ | 4 | 0 | 3.8 | 5 |

available. In this way, we restrict the notion of sampling to refer only to the identification of documents during steps (1) and (4) above, and not for identifying those that should be assessed for relevance before learning can occur.

### *2.1.2 How to sample?*

Different strategies can be used to create the sample. For instance, in the LETOR v3.0 learning to rank test collection (Qin et al. 2009), the top 1,000 documents are sampled using BM25 computed on a document representation that includes the body of the documents and the anchor text of their incoming hyperlinks. Cambazoglu et al. (2010) also used BM25 to create the sample. Liu (2009) does not provide any more details on how the sample should be obtained, but instead, simply states that while using BM25 alone to produce the sample is sufficient, it may not be the best approach. Hence, there are no empirically established guidelines for how to sample, nor for the properties of an effective sample for deploying learning to rank within a search engine. In this work, we investigate the use of anchor text in the documentation representation used by the sample, to determine its impact on the recall of the sample, and the effectiveness of the resulting learned models.

In the earlier LETOR v2.0 test collection, the sample included relevant documents from the pooled TREC relevance assessments, in addition to the top-ranked documents according to BM25. However, Minka and Robertson (2008) noted a bias in this document sampling strategy. In particular, it was observed that a learning to rank technique found a negative feature weight for BM25 to be effective, as this allowed relevant documents not present in the BM25 sample to be easily identified. On the other hand, such a learned model would not be effective on a sample obtained using BM25 alone, as it would rank highly documents with low BM25 scores, despite the fact that these are likely to be irrelevant. To illustrate this *sample selection bias*, Table 1 provides an example learning sample with features, where a selection of top-ranked documents have been identified using the BM25 weighting model within step (1). Two relevant documents (docid 4 & 7) have been added to the sample, in the hope that their presence will permit the learning to rank technique to learn how to rank these documents highly. However, their presence can cause bias within the obtained learned model, which may identify that these relevant documents can be highly ranked simply by identifying documents with the lowest BM25 score. Such a learned model would not perform well on a sample obtained from step (4) using BM25 alone, as it would highly rank documents with the lowest BM25 scores.

Other works in the literature have made use of active learning in the identification of sample during learning (Donmez and Carbonell 2009; Long et al. 2010). By adapting approaches previously used for sampling examples for learning classifiers, it was shown that effective models can be obtained with less documents considered during learning. However, neither of these works examine how the properties of sample are affected, such that the distribution of documents in the training samples remains comparable to that observed in the sample on which the learned model will be applied, to ensure that sample selection bias does not occur. In this work, our realistic setting always ensures that we obtain learned models from training samples that are identified in the same manner as the samples for test queries. For this reason, we do not consider it advantageous to sub-sample the learning samples, such as using active learning approaches, as their impact on the generality of the models remains unclear.

Aslam et al. (2009) compared different strategies for sampling documents, by combining sampling for pooling and sampling for learning [steps (0) and (1)]. In particular, strategies that pooled documents from multiple different retrieval systems (in this case participating systems in TREC 6–8) were found to generate high quality document samples. However, we argue that using multiple retrieval systems to generate the sample for learning may incur a similar bias to that reported by Minka and Robertson (2008) when the learned model is deployed using features obtained from a single system. Moreover, such strategies are not re-usable in a deployed search engine setting, as multiple different systems are not necessarily available with which to generate the sample. In this respect, Chapelle et al. (2011) noted that such approaches from the learning to rank literature only consider *“an offline reranking scenario”*. Naturally, the approach of Aslam et al. (2009) could potentially be achieved using multiple retrieval *models* for generating the sample instead of multiple *systems*. However, in practice, it is the diversity of entirely different retrieval systems rather than different retrieval models that contribute to the quality of a test collection pool (Beitzel et al. 2004; Zobel 1998).

In this work, we aim to learn an effective model for a given search engine. For this reason, we assume a realistic scenario, as identified from the literature, which we summarise as follows: The sample for applying the learned model [step (4)] should be generated by only a single feature (e.g. BM25) from the single system at hand (Chapelle et al. 2011); The sample for learning [step (1)] should have the same number of documents as for applying the learned model and generated using the same method (Liu 2009), to ensure a similar distribution of documents in the training and test samples; We assume that sampling for pooling [step (0)] has occurred independently of the learning to rank process, such that relevance assessments are already available for the documents in the sample, while to avoid sample selection bias, no additional relevant documents are added to the sample (Minka and Robertson 2008).

### *2.1.3 Quality versus quantity*

It is intuitive that the strategy used to create the sample of documents to re-rank will impact on the effectiveness of the learned model. In particular, to maximise the effectiveness of the learned model, the *recall* of the sample should be as high as possible, as this maximises the potential for the learned model to promote the relevant documents to the top of the ranking. To achieve this objective, two alternative strategies are possible: either the approach used to generate the sample should be as effective as possible, or the sample should be large enough to achieve sufficient recall.

The natural way to improve the effectiveness of the sample would be to introduce more features into the ranking process, such that a sample with higher recall for the same number of documents is obtained. However, this induces a recursive application of learning to rank, i.e. learning to rank would be required to generate a sample on which to apply learning to rank.

Instead—as discussed in Sect. 2.1.2—the normal practice is to select many documents ranked by a standard weighting model (e.g. BM25) (Cambazoglu et al. 2010; Qin et al. 2009). By considering a large number of documents for the sample, the recall of the sample is increased, and it is hoped that the learned model can re-rank highly the relevant documents that were not present in the smaller sample. However, larger samples would typically contain more irrelevant documents, with the proportion of newly identified relevant documents compared to irrelevant ones diminishing as the size of the sample increases. For some learning to rank techniques, we postulate that the high imbalance between relevant and irrelevant documents brought by a large sample may hinder the learning of an effective learned model.

In terms of the quality of the document sample, we consider that the choice of the document representation used for obtaining the sample may have an impact upon effectiveness. In particular, adding anchor text to the document representation can alleviate the vocabulary mismatch problem (Plachouras 2006), while also permitting easier identification of the relevant documents for navigational queries (Hawking et al. 2004). However, adding anchor text to the document representation used to generate the sample can significantly change the documents retrieved in the sample, and may not lead to similar improvements for non-navigational queries (Plachouras 2006).

In summary, a smaller, more precise sample could be used, but this boils down to iterative applications of learning to rank. Instead, using a larger, less precise sample leads to the same relevant documents being present as in a smaller, more precise sample. However, a larger sample incurs the expense of further irrelevant documents and decreased efficiency, in terms of deployment and learning times (as illustrated by Fig. 2). Hence, in the next section, we discuss how the size of the sample has been addressed in previous works.

### *2.1.4 How much quantity?*

Assuming that the sample should be generated by a single system using a standard weighting model—as per Sect. 2.1.2—we aim to determine how many documents should be considered in the sample to obtain sufficient recall to allow effective learned models to be obtained.

In general, recall is an area which has seen less work within IR, primarily because precision is deemed the most important aspect in many search applications. Notable exceptions are the types of information needs present within the patent (Piroi and Zenz 2011) and legal search (Tomlinson and Hedin 2011) domains where recall is an overriding important aspect. To this end, models are being developed to determine when to terminate a ranked list while ensuring recall (Arampatzis et al. 2009).

To provide an overview of the sample size used in the literature for various types of information needs, Table 2 quantifies the samples in existing learning to rank test collections. It also reports the type of information needs addressed in each query set, in terms of informational (denoted Inf.), or navigational (Nav.). For some test collections, the particular information need addressed is not specified in the associated literature (Unspec.).

From Table 2 and the literature, we observe that two ranges of sample size are commonly used:

**Table 2** Existing learning to rank test collections

| Test collection | Corpus | | Query set | Information need | Num. queries | Num. features | Ave. sample | Ave. rel. |
|---|---|---|---|---|---|---|---|---|
| | Name | Size | | | | | | |
| LETOR v3.0 | GOV | 1 M | NP03 | Nav. | 150 | 64 | 991.0 | 1.03 |
| | GOV | 1 M | NP04 | Nav. | 75 | 64 | 984.5 | 1.05 |
| | GOV | 1 M | HP03 | Nav. | 150 | 64 | 984.0 | 1.26 |
| | GOV | 1 M | HP04 | Nav. | 75 | 64 | 992.1 | 1.10 |
| | GOV | 1 M | TD03 | Inf. | 50 | 64 | 981.2 | 8.31 |
| | GOV | 1 M | TD04 | Inf. | 75 | 64 | 988.6 | 14.88 |
| | Ohsumed | 350 K | – | Inf. | 106 | 45 | 152.3 | 46.0 |
| LETOR v4.0 | GOV2 | 25 M | MQ2007 | Unspec. | 1,692 | 46 | 41.1 | 12.4 |
| | GOV2 | 25 M | MQ2008 | Unspec. | 784 | 46 | 19.4 | 5.2 |
| MSLR | Web | Unspec. | 10 k | Unspec. | 10,000 | 136 | 120.0 | 59.47 |
| MSLR | Web | Unspec. | 30 k | Unspec. | 30,000 | 136 | 119.6 | 59.9 |
| Yandex IMAT 2009 | Web | Unspec. | – | Unspec. | 9,124 | 245 | 10.6 | 4.9 |
| Yahoo! LTR Challenge | Web | Unspec. | – | Unspec. | 10,871 | 699 | 23.9 | 19.57 |

Unspec. is stated when the relevant literature does not specify the size of the corpus or the information need represented by the test collection

- *Large Samples* ($\geq$ 1,000 *documents*): Samples of 1,000 documents are used by the early LETOR test collections (v3.0 and before) (Qin et al. 2009). Indeed, for the TREC GOV collection, it was previously claimed that 1,000 documents are sufficient for combining BM25 and PageRank, without loss of effectiveness (Craswell et al. 2005). Craswell et al. (2010) used a sample of 5,000 documents for retrieval from the 500 million documents of the ClueWeb09 collection, but without justification. Chapelle et al. (2011) anecdotally report sample sizes of *"tens of thousands of documents"*.
- *Small Samples* ($\leq$ 200 *documents*): From Table 2, we observe that various learning to rank test collections have mean sample sizes of 120 documents or less (e.g. LETOR v4.0, MSLR), with some as few as 10 documents per query (Yandex), even when sampled from the entire Web. Similarly, Zhang et al. (2009) used a sample of 200 documents ranked by BM25 and PageRank from a Web search engine.

It is noticeable that while for many of the query sets the type of information need addressed is not specified, we expect these to include both information and navigational needs, perhaps for the same query. Indeed, similar to recent TREC Web track test collections (Clarke et al. 2010, 2011), there is a trend towards using multiple relevance label grades to address different possible information needs for the same query. For example, for the query 'University of Glasgow', the University's home page would be labelled 4 ('perfect'), representing the best answer for the navigational information need of a typical user, while the Wikipedia entry about the University would be labelled 3 ('excellent'). Documents not discussing the University would be judged irrelevant (labelled 0). The high mean numbers of relevant documents for query sets such as MSLR, Yandex and Yahoo! LTR Challenge in Table 2 are indicative that labelling with multiple relevance grades is used for these datasets. However, we observe no relationship between the type of information need and the size of the sample used within the existing learning to rank test collections.

Overall, there is clearly a lack of evidence on the appropriate size of the document sample for effective learning. Hence, in this work, as the first of three research themes, we formulate and validate several hypotheses relating to the sample, allowing us to empirically determine the properties of an effective sample. Firstly, we expect that, in general, sample size does have an impact on effectiveness:

**Hypothesis 1** The observed effectiveness of learned models can be affected by different sample sizes.

Next, depending on the presence of anchor text within the document representation used for the sample, the few relevant documents for navigational information needs may be easier to find than the larger number of relevant documents for informational information needs. In particular, the choice of document representation to use when sampling will likely impact on the documents identified in the sample. Moreover, the most suitable document representation for sampling may vary across different types of information needs. For instance, if the document representation used by the weighting model for obtaining the sample does consider anchor text, then the navigational pages with quality anchor text in their incoming hyperlinks are more likely to be retrieved in the sample (Hawking et al. 2004; Plachouras and Ounis 2004). However, we postulate that using anchor text may reduce the number of relevant documents identified for more informational queries. Hence, we hypothesise that:

**Hypothesis 2** The observed effectiveness of learned models can be affected by the type of information need observed in the queries, and the used document representation for generating the samples, regardless of the size of these samples.

Moreover, the choice of learning to rank technique may also have an impact on the effective choice of the sample size. For instance, pairwise learning to rank techniques aim to reduce the number of incorrectly ranked pairs of documents. However, as larger sample sizes exhibit less balance between relevant and irrelevant documents, there are larger number of document pairs for which no preference relation exists, which may degrade the effectiveness of these learning to rank techniques. Hence, we hypothesise that:

**Hypothesis 3** The observed effectiveness of learned models depends on the deployed learning to rank technique and the sample size.

Finally, as discussed above in Sect. 2.1.1, minimising the size of the sample without significant degradations compared to an *effective sample size* has marked efficiency benefits. In particular, when applying the learned model, minimising the number of sample documents to be identified during matching reduces the response time, while also reducing the time required to calculate features for all documents in the sample. Moreover, smaller sample sizes reduce the learning time for most learning to rank techniques (see Fig. 2). For these reasons, the identification of the smallest sample size for obtaining effective retrieval performance is desirable, which we aim to answer in the following research question:

**Research Question 1** What are the aspects that define the smallest sample size for an effective learned model?

### 2.2 Learning evaluation measure

Besides investigating the properties of an effective sample, we are also interested in the effective configuration of learning to rank techniques. Indeed, for listwise learning to rank techniques, the evaluation measure deployed during learning can impact on the effectiveness of the learned models (Xu and Li 2007). In the following, we review and discuss the use (Sect. 2.2.1), properties (Sect. 2.2.2) and choice of evaluation measures during learning (Sect. 2.2.3). This discussion is used to motivate hypotheses concerning our second and third research themes, which address the role of the learning evaluation measure, and the effect of the interaction between the learning evaluation measure cutoff and the sample size.

### *2.2.1 Need for evaluation measures in learning*

In the process of learning a model, a learning to rank technique will attempt many different combinations of features, and evaluate them as per the defined loss function. *Listwise* learning to rank techniques, which directly deploy IR evaluation measures as their loss function [e.g. AFS (Metzler 2007) and AdaRank (Xu and Li 2007)] are reported to be particularly effective (Liu 2009).

Many evaluation measures have been defined in the literature, with different properties. However, their role within learning has not been the subject of an in-depth investigation. Indeed, in a keynote presentation, Croft (2008) speculated that different optimisations might be needed for different measures. In the following, we discuss the attributes of evaluation measures, and their impact when a measure is used for learning.

### *2.2.2 Properties of evaluation measures*

In general, IR evaluation measures—such as Mean Average Precision (MAP) (Buckley and Voorhees 2000) and normalised Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen 2002)— are not continuous. This is because each measure may or may not ‘react’ to a swap in the positions of two documents, depending on the relevance of the two documents and their respective positions in the ranked list. Moreover, measures are computed to a pre-determined *cutoff* rank, after which they do not consider the retrieved documents. For instance, precision only considers the number of relevant documents occurring before the cutoff, but not the order of the documents before the cutoff, nor the order after.

The responsiveness of a measure to swaps in the ranking is known as its *informativeness* (Robertson 2008). For example, consider a ranking of four documents for a query, of which one is highly relevant, two relevant, and one irrelevant. Of the $4! = 24$ possible rankings of these four documents, precision can only give one possible value (0.75), and mean reciprocal rank (MRR) can give two (1 or 0.5). In contrast, MAP and NDCG are more informative measures, as they discriminate between the effectiveness of more of the 24 possible rankings, producing four and nine different values, respectively.

### *2.2.3 Evaluation measures for effective learning*

Given a measure on the test queries that we wish to improve, it is not necessarily the case that we should also aim to maximise this measure during the learning process (He et al. 2008; Robertson 2008). Indeed, the informativeness of the learning measure will have a direct impact on the learned model, and hence a measure used for testing may not be sufficiently informative to guide the learning process. For instance, as precision at *k* does not react to many swaps in the document ranking during learning, it may not be able to differentiate between two possible models whose performance characteristics are actually quite different (e.g. consider two models that place a highly relevant document ranked at rank 1 and rank *k*, respectively). In contrast, when graded relevance assessments are available, NDCG and ERR are more informative measures, as they can differentiate between the relevant documents of different relevance grades (e.g. consider the ordering of a pair of documents, one highly relevant, one marginally relevant).

Next, the rank cutoff also affects the usefulness of the measure. For instance, by only considering documents to rank *k*, a measure cannot differentiate between ranking models that place a relevant document at rank $k + 1$ or rank $k + 100$—even if the former is more likely to be an effective model on unseen data (Robertson 2008). However, for a given sample of documents, using larger cutoffs during learning also degrades the efficiency of the learning to rank technique. For example, in Fig. 2, it takes AFS using NDCG@1,000 approximately twice as long to learn a model as AFS using NDCG@10 (e.g. 2,097 vs. 6,097 s for 5,000 document samples). Moreover, as most IR evaluation measures are ‘top-heavy’, they react less to changes deeper in the ranking, meaning that we expect diminishing returns in the effectiveness of the resulting model as the evaluation cutoff is increased.

The impact of the choice of measure and cutoff in listwise learning to rank techniques have not been the subject of much literature. An exception is the work of He et al. (2008), who compared the sensitivity of different measures for training under different levels of incompleteness. However, the training was conducted in the setting of hyper-parameters of retrieval models (e.g. the *b* hyper-parameter of BM25), rather than the now more typical learning to rank environment. Robertson (2008) speculated on the effect of the informativeness of the training measure, without however providing empirical evidence. Donmez et al. (2009) found that given enough training queries, for a given test measure, the same measure should be the most suitable for learning. Later, Yilmaz and Robertson (2010) challenged this finding, noting that sufficient queries are not always available. Our work further aids the understanding of this area, by investigating the choice of the learning evaluation measure within a learning to rank setting, across multiple search tasks, and with even sparser amounts of training queries.

As our second research theme, we formulate several hypotheses relating to the choice of the learning evaluation measure within a listwise learning to rank technique [step (3)]. Firstly, based on the different informativeness of various evaluation measures, we hypothesise the following:

**Hypothesis 4** The observed effectiveness of the learned model obtained from a listwise learning to rank technique can be affected by the choice of the learning evaluation measure.

For a large sample, increasing the rank cutoff of the learning evaluation measure increases the informativeness of the measure by allowing it to potentially respond to more swaps between relevant and irrelevant documents. It follows that by using a measure with improved informativeness, a learning to rank technique may identify a yet unseen feature combination that works well for some pairs of document in the sample, which is not observed for the smaller rank cutoff. This may be useful later for improving the effectiveness of an unseen query, where the relevant documents are pushed to the top ranks, due to the additional feature combination. Hence, we hypothesise that:

**Hypothesis 5** The observed effectiveness of the learned model obtained from a listwise learning to rank technique can be affected by the choice of the rank cutoff of the learning evaluation measure.

Lastly, as our third and final research theme, we combine the two investigations within this work concerning the sample size [step (1) and (4)] and the rank cutoff of the selected learning evaluation measure [step (3)], to investigate the dependence between the two. Consider the evaluation of 4 documents discussed above. If the sample is also of 4 documents, then there are only $4! = 24$ possible rankings that can be produced. However, if the sample size is larger, say 5, then there are $5! = 120$ possible rankings, even though the evaluation measure will only consider the ordering of the top-ranked 4 documents. The size of the increased space for the possible re-ranking should increase the chances that an effective learned model can be identified. For this reason, our last hypothesis is the following:

**Hypothesis 6** The observed effectiveness of the learned model obtained from a listwise learning to rank technique can be affected by both the choice of the rank cutoff of the learning evaluation measure and the size of the samples.

In the following, we define the experimental methodology (Sect. 3), as well as the experimental setup (Sect. 4) permitting the investigation of our three research themes. Recall that these three research themes address the role of the sample, the role of the learning evaluation measure, and how the sample size and the learning evaluation measure interact, respectively. Experimental results are reported and discussed in Sect. 5.

## 3 Methodology

In this section, we define the methodology that permits the identification of best practices for the three research themes mentioned above. As discussed in Sect. 2.1.2, we assume that the learned model will be deployed on a single retrieval system, with the sample generated by using a standard weighting model. For learning, documents in the sample are labelled using high quality relevance assessments (e.g. from TREC) that are already available.

We investigate the research questions and hypotheses of our three research themes across different scenarios, including different types of information needs, learning to rank techniques, and corpora. To investigate each of our research questions, we perform empirical experiments using multiple query sets on several learning to rank test collections. In particular, query sets are grouped into training, validation and testing sets with no overlap of queries. For instance, for a given query set (HP04, NP04 etc.), LETOR v3.0 GOV prescribes five folds, each with separate training, validation and testing query sets. On a given testing query set, the test evaluation measure and its rank cutoff are pre-defined—to use the experimental design terminology, this is the *dependent variable*. Then, on the corresponding training query set, the “factor of interest” (*independent variable*) for a particular learning to rank technique is varied (namely sample size, or evaluation measure and cutoff). For each setting of a learning to rank technique (e.g. sample size, learning evaluation measure and cutoff), a model is learned. The effectiveness of this learned model is then evaluated on the test query set using the test evaluation measure. By comparing across different learning settings, the impact of the factor being varied can be assessed.

Within this framework, the columns of Table 3 detail the experimental design for each of our three research themes on a given pair of training/testing query sets. In particular, to investigate the role of the sample size, the *original* samples for the train and test query sets are both cropped to various different sizes, while maintaining their ordering of documents. Moreover, as discussed in Sect. 2.1.2, we ensure that a learned model is not deployed on a sample that has a different size from that with which it was learned. Using this methodology, we can vary the sample size as the primary factor in the experiment, thereby

**Table 3** How experimental variables are varied or fixed for each research question on a given query set

| Experimental variables | Type | Research themes | | |
|---|---|---|---|---|
| | | Sample size | Learning measure & cutoff | Learning cutoff & sample size |
| Train & test sample size | Independent | 1st Factor | Fixed | 1st Factor |
| Learning measure | Independent | Fixed | 1st Factor | Fixed |
| Learning measure cutoff | Independent | Fixed | 1st Factor | 1st Factor |
| Learning to rank technique | Independent | 2nd Factor | 2nd Factor | Fixed |
| Sample document representation | Independent | Other factor | Fixed | Fixed |
| Test measure | Dependent | Fixed | Fixed | Fixed |

simulating the smaller samples used by other learning to rank test collections. However, the learning evaluation measure and cutoff (only applicable for listwise learning to rank techniques) as well as the test evaluation measure are held fixed. In this way, we can observe the impact of sample size on the effectiveness of the learned model. Moreover, as the conclusion may change for different learning to rank techniques or depending on the document representation used by the sampling strategy (i.e. anchor text or not), we vary these as additional "second" or "other" factors of interest.

To investigate our second research theme concerning the role of the learning evaluation measure for listwise learning to rank techniques, following Table 3, we firstly fix the sample size, as well as the evaluation measure for testing. The primary factors in this experiment are the learning evaluation measure and the rank cutoff—we vary both of these concurrently, to allow any dependent effects between the choice of the learning evaluation measure and the rank cutoff to be examined. Furthermore, as a second experimental factor, this experiment is conducted for different listwise techniques, to examine if the conclusions change for different techniques.

Lastly, for our third research theme, we vary both the sample size and the learning measure cutoff within a single experiment, to determine if there is any *dependence* between these factors. Hence, as shown in Table 3, both the sample size and the learning evaluation measure rank cutoff form the (1st factor) independent variables of interest.

For each set of experiments, to measure if there is a significant dependence between effectiveness and the first or second factors, we use within-subject ANOVAs (Coolican 1999) to measure the probability that the observed results are caused by the null hypothesis for each factor. In particular, each query is a subject, where the dependent variable (the test evaluation measure) is observed after each variation of the independent variables. Indeed, by using ANOVAs, we can measure that the observed variance across the different independent variables is not due to a type I error.

## 4 Experimental setup

Our experiments across the three research themes are conducted using the methodology defined in Sect. 3, and are reported in Sect. 5. In this section, we describe the setting of these experiments, structured in three subsections: Sect. 4.1 describes the learning to rank test collections used in our experiments, Sect. 4.2 provides an overview of the used learning to rank techniques, and Sect. 4.3 describes how we vary the experimental setup for the various factors identified in Sect. 3.

### 4.1 Learning to rank test collections

As detailed in Sect. 3, we experiment with document samples of various sizes. To this end, the learning to rank test collections that we apply must have sufficiently large original samples to allow cropping into smaller samples.

Of all learning test collections in Table 2, we consider that only the LETOR v3.0 GOV query sets have sufficiently large samples [$\sim$1,000 documents per query (Qin et al. 2009)] to allow smaller sample sizes to be investigated. Each of the six query sets is split into fivefolds for the purposes of cross validation (train, validation and test), and have binary relevance assessments as well as a common set of 64 features. We select three query sets from TREC 2004, covering different types of information needs with the same number of queries: HP04 (home page) and NP04 (named page) query sets both represent navigational information needs with single relevant documents; while TD04 (topic distillation) is an informational query set.[1]

However, as discussed in Sect. 2.1.4, the LETOR v3.0 GOV query sets do not exhibit the largest sample sizes seen in the literature [5,000 documents per query sampled from the ClueWeb09 collection (Craswell et al. 2010)]. To investigate whether such large samples are necessary, and to strengthen our conclusions, we create a new, larger and more modern learning to rank test collection. In particular, the TREC 2009 Million query track has various queries sampled from the logs of a Web search engine, while the TREC 2009 & 2010 Web tracks addressed queries that are faceted or ambiguous in nature (which can mix informational and navigational interpretations for each query), also sampled from the logs of a Web search engine. We select three such *mixed* query sets of queries from these recent TREC tracks that include graded relevance assessments:

- *MQ09:* 70 queries from the TREC 2009 Million query track (Carterette et al. 2010).
- *WT09:* 50 queries from the TREC 2009 Web track (Clarke et al. 2010).
- *WT10:* 48 queries from the TREC 2010 Web track (Clarke et al. 2011).

Lastly, we use a query log to create a purely navigational query set, so that we can directly compare and contrast results with the purely navigational HP04 and NP04 query sets from LETOR v3.0:

- *NAV06:* 150 'head' queries with the highest clickthrough into ClueWeb09 documents as suggested by Macdonald and Ounis (2009), obtained from the MSN 2006 query log (Craswell et al. 2009).

The underlying document corpus for the MQ09, WT09, WT10 and NAV06 query sets is the TREC 'category B' ClueWeb09 Web crawl (CW09B). This corpus comprises 50 million English documents, aimed at representing the first tier of a commercial search engine index. We index these documents using the Terrier retrieval system (Ounis et al. 2006), including anchor text from incoming hyperlinks. For each query, we use a light version of Porter's English stemmer and the DPH (Amati et al. 2008) Divergence from Randomness weighting model to extract large original document samples for each query set. As we are not aware of the particular parameter settings used for BM25 when sampling in the LETOR v3.0 query sets, using DPH permits comparable sample effectiveness to BM25, without the need to train any parameter, as DPH is parameter free (Amati et al. 2008).

To cover a wide range of sample sizes, we aim to sample as many documents as possible from CW09B for each query. However, as noted in Sect. 2.1.1, the learning time

[1] We collectively refer to HP04, NP04 and TD04 as the GOV query sets.

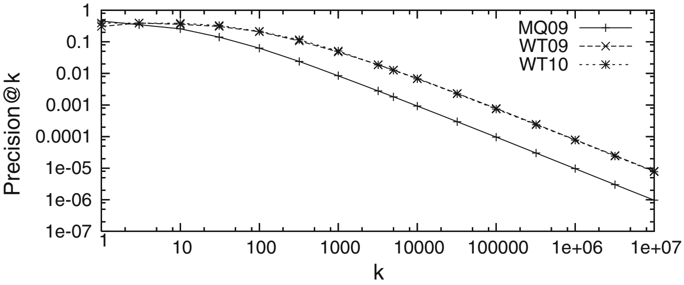


**Fig. 3** Precision@k for the CW09B query sets. Lowest ranked relevant document found at rank 3.9 × $10^6$. WT09 & WT10 are super-imposed

of various learning to rank techniques increases with sample size. To keep our experiments feasible, we investigate the number of relevant documents identified for the query sets when the number of retrieved documents is unconstrained. Figure 3 reports the precision@*k*, for *k* up to the size of the corpus (50 million documents). We note that after rank $k = 5{,}000$, precision falls below 0.01, meaning that, on average, for every additional 100 documents retrieved, at most 1 more relevant document will be retrieved. Moreover, from Fig. 2, running AFS for the WT09 query set using 5,000 documents can take over 1.5 hours. For these reasons, we deem an original sample of 5,000 documents to be sufficient for these experiments. Indeed, on a detailed inspection of the experiments conducted for Fig. 3, we find that all three query sets have 80–90 % recall at 5,000 documents compared to an exhaustive sampling comprising the union of all documents containing any query term.

Next, recall that our experiments consider the role of anchor text in the document representation used to identify the sample and its impact on retrieval effectiveness for different types of informational need. We firstly highlight the document representations that we consider. Figure 4 illustrates the fields of an example document, in terms of the title, body, URL and anchor text. In the GOV query sets, the documents in the sample are obtained by computing BM25 on all fields, including anchor text. To facilitate analysing the impact of the document representation used in the sampling, the document samples identified using DPH on the CW09B query sets are created using two different document representations, without and with the presence of anchor text. When sampling with anchor text, the ranking of documents in the sample changes, making it more likely that the homepage documents with much anchor text will be highly ranked for a corresponding navigational query, such as the query ‘Bing’ for the example document in Fig. 4.

For the CW09B query sets, we calculate a total of 75 document features for all documents in the samples, covering most of the features used in the learning to rank literature (Qin et al. 2009). These features are summarised in Table 4 and organised into the following five classes:

- *Standard weighting models (WM)* computed on each field of each document, namely title, body, anchor-text and URL. We use four weighting models, namely BM25 (Robertson et al. 1992), PL2 (Amati 2003), DPH (Amati et al. 2008) and LM with Dirichlet smoothing (Zhai and Lafferty 2001). We note that the chosen document representation for generating the sample does not have an impact on the availability of the four fields—for instance, we may use a document representation without anchor text for generating the sample, but we can still calculate WM features based on the anchor text.

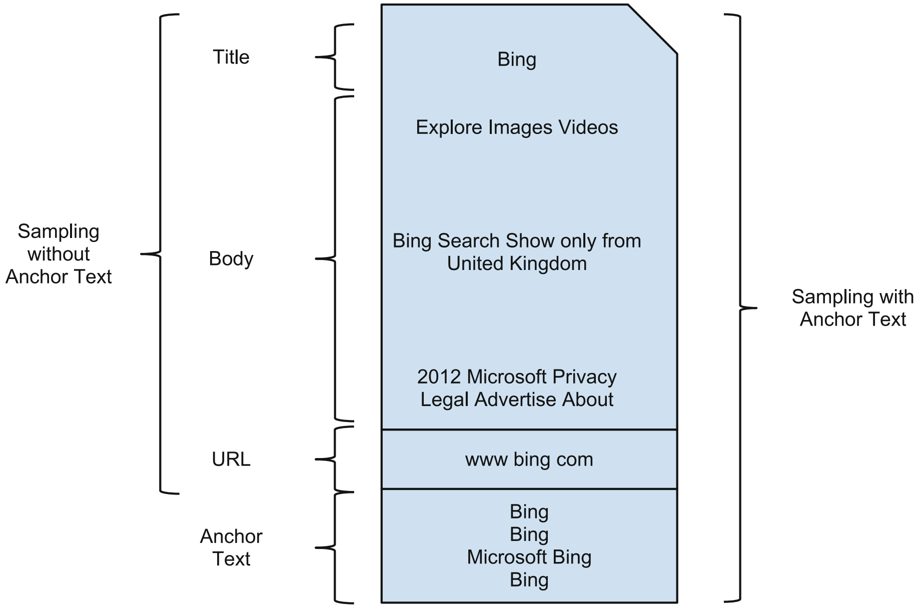

Sampling without Anchor Text
Title
Body
URL
Anchor Text
Bing
Explore Images Videos
Bing Search Show only from United Kingdom
2012 Microsoft Privacy Legal Advertise About
www bing com
Bing
Bing
Microsoft Bing
Bing
Sampling with Anchor Text

**Fig. 4** Document representation used within our work

**Table 4** Features deployed for the three query sets of our new CW09B learning to rank test collection

| Class | Feature | Description | Total |
|---|---|---|---|
| LA | Absorbing | Absorbing model score (Plachouras et al. 2005) | 2 |
| LA | Edgerecip | No. of reciprocal links (Becchetti et al. 2006) | 2 |
| LA | Inlinks | No. of inlinks | 2 |
| LA | Outlinks | No. of outlinks | 2 |
| LA | InvPageRank | PageRank transposed score | 2 |
| LA | PageRank | PageRank score (Page et al. 1998) | 2 |
| SPAM | SpamFusion | Spam likelihood (Cormack et al. 2011) | 2 |
| URL | URLDigits | No. of digits in domain and host | 4 |
| URL | URLComps | No. of host, path, and query components | 6 |
| URL | URLLength | Length of host, path, and query string | 6 |
| URL | URLType | Root, subroot, path, file (Kraaij et al. 2002) | 2 |
| URL | URLWiki | Whether URL is from Wikipedia | 2 |
| WM | BM25 | BM25 score (Robertson et al. 1992) | 5 |
| WM | DPH | DPH score (Amati et al. 2008) | 5 |
| WM | LM | LM score (Dirichlet) (Zhai and Lafferty 2001) | 5 |
| WM | PL2 | PL2 score (Amati 2003) | 5 |
| WM | MQT | No. of matching query terms | 5 |
| WMP | MRF | MRF dependence score (Metzler and Croft 2005) | 8 |
| WMP | pBiL | DFR dependence score (Peng et al. 2007) | 8 |
| TOTAL | | | 75 |

- *Link analysis-based features (LA)* typically identify authoritative documents. We deploy PageRank and incoming and outgoing link counts.
- *Proximity weighting models (WMP)* boost documents where the query terms occur in close proximity. For each field, we deploy proximity language models (Metzler and Croft 2005) and DFR models (Peng et al. 2007) based on Markov Random Fields.
- *URL features (URL)* (e.g. short URLs) are often a useful feature for identifying homepages (Kraaij et al. 2002).
- *Spam* documents are present in the realistic CW09B corpus. We include the fusion score by (Cormack et al. 2011) as a feature.

All parameters of the document features (e.g. $b$ in BM25, $c$ in PL2, $\mu$ in LM with Dirichlet smoothing) remain at their recommended settings, as implemented by the Terrier retrieval platform, namely $b = 0.75$, $c = 1$, and $\mu = 2500$. Finally, as different features are measured on different scales, we follow the common practice (Liu 2009) [and in line with LETOR (Qin et al. 2009)] to normalise all features to lie between 0 and 1 for each query.

### 4.2 Learning to rank techniques

Various learning to rank approaches in the literature fall into one of three categories, namely pointwise, pairwise and listwise (Liu 2009). In this work, we deploy learning to rank techniques representative of each of the three categories, all of which are either freely available as open source or widely implemented:

- *GBRT (Pointwise)*, also known as Gradient Boosted Regression Trees, produces a set of regression trees that aim to predict the label of documents in the training data (Friedman 2000). The tree learned at each iteration only needs to find the difference between the target label and the prediction of the previous tree(s). We use the RT-Rank implementation (Weinberger et al. 2010).[2] However, going further than the open source implementation, we choose the number of trees that performs highest on the validation data as the final learned model.
- *RankBoost (Pairwise)* constructs a linear combination of weak rankers (in our case the various document features), based on a loss function defined as the exponential difference between the labels of pairs of documents (Freund et al. 2003).
- *RankNet (Pairwise)* constructs a neural network, based on a loss function encapsulating the cross entropy of pairs of objects being correctly ranked (Burges et al. 2005).
- *LambdaMART (Pairwise/Listwise[3])* also deploys boosted regression trees internally, but the training of the trees consider NDCG[4] to obtain the gradient of the surrogate loss function between pairs of documents (Wu et al. 2008). We use the implementation of the Jforests open source package (Ganjisaffar et al. 2011).[5] A LambdaMART approach was the winning entry in the Yahoo! learning to rank challenge (Chapelle and Chang 2011). The model of the highest performing iteration on the validation data is chosen.
- *AFS (Listwise)*, also known as Automatic Feature Selection, obtains a weight for the linear combination of the most effective feature at each iteration, which is then added

[2] http://sites.google.com/site/rtranking/.

[3] LambdaMART is both pairwise and listwise according to Li (2011).

[4] It is not trivial to change LambdaMART to arbitrary learning measures.

[5] http://code.google.com/p/jforests/.

to the set of features selected in the previous iteration(s) (Metzler 2007). In our implementation, we use simulated annealing (Kirkpatrick et al. 1983) to find the combination weight for each feature that maximise NDCG@1,000. Note that such weights are obtained one by one, with no retraining of the weights of those already selected features. When validation data is used, the model of the highest performing iteration as measured using the same evaluation measure on the validation data is chosen [in this manner, the validation data is used to determine the correct number of AFS iterations, as suggested by Liu (2009)].

- *AdaRank (Listwise)* optimises feature weights by applying boosting (Xu and Li 2007). In particular, at each iteration, a distribution of the importance of each query is updated, and the weight of the feature that improves the overall performance of those queries, after weighting by the importance distribution, is added to the model. A feature can be selected multiple times and its weight consequently updated. As suggested by Liu (2009), we use validation data to set the number of iterations, so as to prevent the overfitting of models on the training set.

### 4.3 Experimental factors

In order to address the hypotheses and research questions identified in Sects. 2.1.4 and 2.2.3, we analyse the outcome of various learning to rank settings on a corresponding test query set. Table 5 details how the query sets are used for training, validation and testing for the learning to rank test collection based on the GOV and CW09B corpora. In particular, for the GOV query sets, we use the fivefolds prescribed in LETOR v3.0, including the splits of the topics into separate training, validation and testing sets for each fold; we report the mean over all of the test topics from all folds. For the CW09B corpus, we retain a setting as realistic to a TREC deployment as possible. In particular, for WT10, we use the query set from the preceding TREC year (i.e. WT09) for training (60 %) and validation (40 %). As no training queries were available for WT09, we use queries from the TREC 2009 Million Query track (MQ09) for training (60 %) and validation (40 %). Similarly, we split the 150 NAV06 queries into equal sets, to form a single fold with separate training, validation and testing query subsets.

Table 5 also records the test evaluation measure used to test the effectiveness of the learned models on each query set. Indeed, for the GOV query sets, the measure used for each query set matches the official measure used by the corresponding TREC Web track (Craswell and Hawking 2004). For the CW09B query sets, NDCG@20 was used for the evaluation measure in TREC 2009 (Clarke et al. 2010), while for TREC 2010, ERR@20

**Table 5** Applied training, validation and testing query sets

| Task type | Corpus | Folds | Query sets | | | Test measure |
|---|---|---|---|---|---|---|
| | | | Training | Validation | Test | |
| Navigational | GOV | 5 | HP04 | HP04 | HP04 | MRR |
| Navigational | GOV | 5 | NP04 | NP04 | NP04 | MRR |
| Informational | GOV | 5 | TD04 | TD04 | TD04 | MAP |
| Mixed | CW09B | 1 | MQ09 | MQ09 | WT09 | NDCG@20 |
| Mixed | CW09B | 1 | WT09 | WT09 | WT10 | ERR@20 |
| Navigational | CW09B | 1 | NAV06 | NAV06 | NAV06 | MRR |

was used (Clarke et al. 2011). In the following experiments, we use both measures, and for completeness, we additionally use MAP for both WT09 and WT10 query sets. For the NAV06 navigational query set, we report mean reciprocal rank (MRR).

Aside from the deployed learning to rank technique, there are four factors in our experiments that we defined in Sect. 3, namely: the size of the sample; the document representation used to generate the sample; the learning evaluation measure using within the loss function of the AFS and AdaRank listwise learning to rank techniques; and the rank cutoff of the learning evaluation measure. Indeed, as per the methodology prescribed in Sect. 3, we differentiate between the test evaluation measure, which remains fixed, and the learning evaluation measure used by a listwise learning to rank technique, which we vary. The settings for each of the factors in our experiments are as follows:

- *Sample Document Representation*—For GOV, BM25 with anchor text, as provided by LETOR v3.0. For CW09B, DPH with or without anchor text, as illustrated in Fig. 4.
- *Sample Size*—We experiment with different sample sizes, up to the maximum permitted by the original samples of size 1,000 for GOV and 5,000 for CW09B:

$$\begin{aligned}\text{GOV} &= \{10, 20, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1,000\};\\ \text{CW09B} &= \{10, 20, 50, 100, 500, 1,000, 1,500, 2,000, 3,000, 4,000, 5,000.\}\end{aligned}$$

- *Learning Evaluation Measures*—We experiment with a selection of standard measures for use within the loss functions of the listwise learning to rank techniques, which may be different from the test evaluation measure used to assess the retrieval performance of the resulting learned model: Precision (P), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), normalised Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen 2002) and Expected Reciprocal Rank (ERR) (Chapelle et al. 2009).
- *Learning Evaluation Measure Cutoffs*—We also vary the rank cutoff of the learning evaluation measures. The cutoff values used for both GOV and CW09B are the same values used for the sample size, as follows:

$$\begin{aligned}\text{GOV} &= \{10, 20, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1,000\};\\ \text{CW09B} &= \{10, 20, 50, 100, 500, 1,000, 1,500, 2,000, 3,000, 4,000, 5,000\}.\end{aligned}$$

Our chosen sample sizes (and cutoffs) cover small and large sizes observed in existing learning to rank test collections and in the literature. For the learning evaluation measures, we include both standard measures such as precision and MAP, in addition to measures that consider graded relevance judgements (NDCG and ERR). Moreover, due to the interaction of the sample size with the learning evaluation measure cutoff, the learning settings that would produce identical results are omitted. For example, if sample size is 500, MAP to rank 1,000 (i.e. MAP@1,000) is identical to MAP@500. All experimental factors, along with the deployed learners, are summarised for both GOV and CW09B in Table 6.

## 5 Results

In the following, we experiment to address each of our three research themes: Sect. 5.1 addresses the theme concerning the properties of the sample; Sect. 5.2 addresses the learning evaluation measure and cutoff research theme; Sect. 5.3 analyses the final research

**Table 6** All factors in our experiments

| | GOV | CW09B |
|---|---|---|
| Learners | GBRT, RankBoost, RankNet, LambdaMART, AFS, AdaRank | |
| Sample document model & representation | BM25 with anchor text | DPH with/without anchor text |
| Sample size | {10, 20, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1,000} | {10, 20, 50, 100, 500, 1,000, 1,500, 2,000, 3,000, 4,000, 5,000} |
| Learning evaluation measure | P,MAP,MRR,NDCG,ERR | |
| Learning evaluation measure cutoffs | {10, 20, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1,000} | {10, 20, 50, 100, 500, 1,000, 1,500, 2,000, 3,000, 4,000, 5,000} |

theme concerning the dependence between the sample size and the learning evaluation measure cutoff, as well as the features selected in the obtained learned models.

## 5.1 Sample size

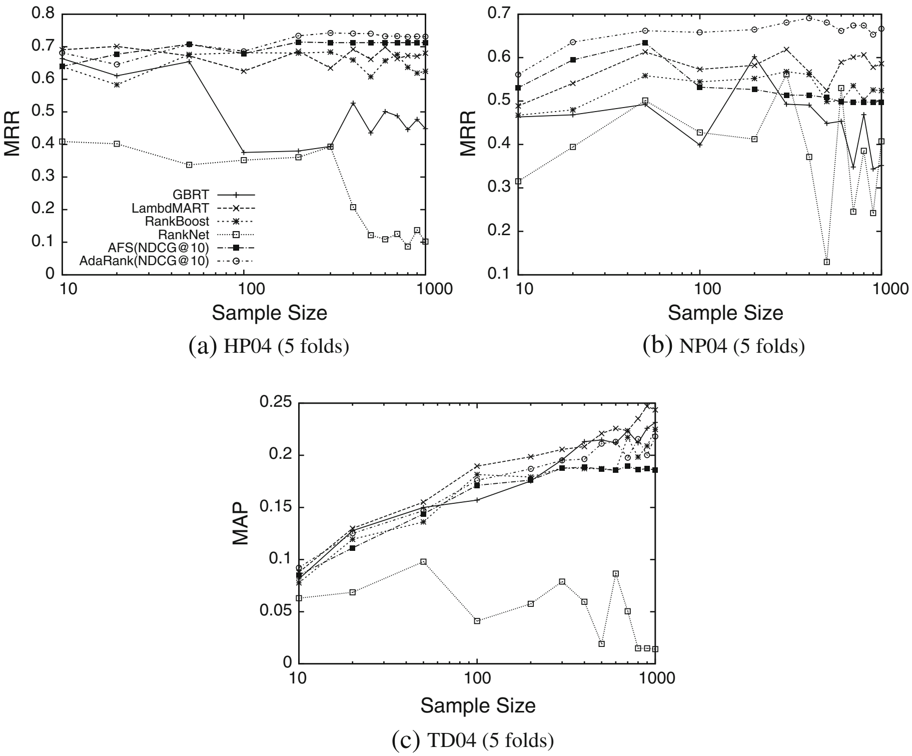
Figures 5, 6 and 7 show the impact of different sample sizes on the test performance, for the GOV (Fig. 5), as well as the CW09B query sets, without and with the use of anchor text when generating the sample (Figs. 6 & 7, respectively). In each figure, the test performance of a learning to rank technique is represented by a line. For the listwise techniques—based on insights that we will discuss in Sect. 5.2—in this section, we fix the learning measure and rank cutoff to NDCG@10.

We firstly make some general observations from Figs. 5, 6 and 7. With respect to sample size, a common trend can be observed across all figures: retrieval performance generally increases as the sample size increases, but stabilises after a sufficiently large sample. However, the size at which effectiveness stabilises can vary widely: for some query sets, all but the smallest sample sizes appear to be effective-indeed, for the HP04 query set (Fig. 5a), even 10 documents appears to be very effective; yet for other query sets, effectiveness continues to rise until much larger samples—e.g. 1,000 in Fig. 6d. At the extreme end, we note that for the NAV06 query set without anchor text (Fig. 6g), a marked rise in effectiveness occurs when sample size reaches 5,000. Moreover, across all query sets, the trends exhibited by the RankNet learning to rank technique represent outliers, which we discuss further below.

In the following, we expand upon this analysis, by examining each of the query sets and learning to rank techniques in turn (Sects. 5.1.1 and 5.1.2), before concluding on each of our defined hypotheses in Sect. 5.1.3.

### *5.1.1 Query set analysis*

Taking each query set in turn, we make observations about the relationship between sample size and effectiveness across these different query sets. Firstly, for the navigational query sets HP04 and NP04 (Figs. 5a, b), we note that a reasonable performance is obtained at small sample sizes, suggesting that most of the relevant documents that can be ranked highly by the learned models have been obtained within a sample of that size. In particular, the LambdaMART learning to rank technique achieves its highest HP04 MRR at sample



(a) HP04 (5 folds)

(b) NP04 (5 folds)

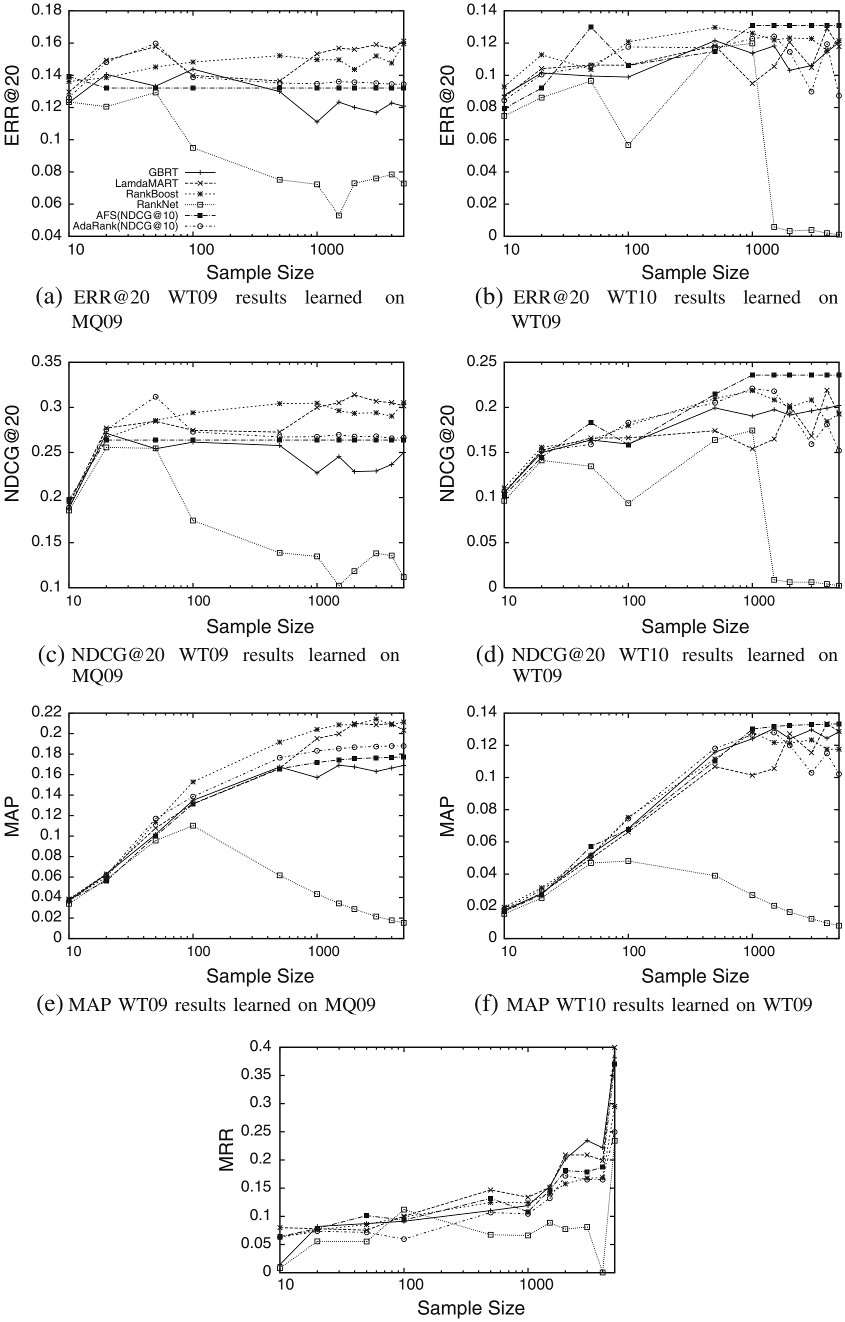
(c) TD04 (5 folds)

**Fig. 5** Effect on test performance (in terms of MRR or MAP) on the LETOR GOV query sets of different sample sizes, learned using various learning to rank techniques. Key is common to **a**–**c**

size 10, while other techniques are all competitive from size 50. On the NP04 query set, sample sizes of 50 or larger are most effective. In contrast, for topic distillation TD04 (Fig. 5c), samples up to 400–600 documents are required before the most effective models are obtained. This larger sample size is expected, as topic distillation represents information needs that are more informational in nature, and suggests that relevant documents may appear at lower ranks in the samples for this query set.

The results for the ClueWeb09 query sets (WT09, WT10 and NAV06) are reported twice: in Fig. 6, documents are sampled using a document representation that does not consider anchor text; in Fig. 7, the sampling document representation includes anchor text in addition to the title, URL and body fields. Moreover, the results for WT09 and WT10 query sets are reported for each of the three test evaluation measures, namely ERR@20, NDCG@20 and MAP. Indeed, while the used evaluation measure changed between the TREC 2009 and TREC 2010 Web tracks from NDCG@20 to ERR@20 (see Table 5), we find that the choice of measure for testing impacts on the obtained conclusions, hence we provide figures for both, while adding the classical MAP measure permits observations about how sample size impacts effectiveness at deeper ranks.

Overall, we note that smaller samples result in lower effectiveness for the ClueWeb09 query sets. However, the degradation observed is dependent on the test evaluation measure being used. Indeed, for ERR@20, an effective model is obtainable with a sample size of 20–50 documents. However, for NDCG@20, in general, maximal effectiveness is not



(a) ERR@20 WT09 results learned on MQ09

(b) ERR@20 WT10 results learned on WT09

(c) NDCG@20 WT09 results learned on MQ09

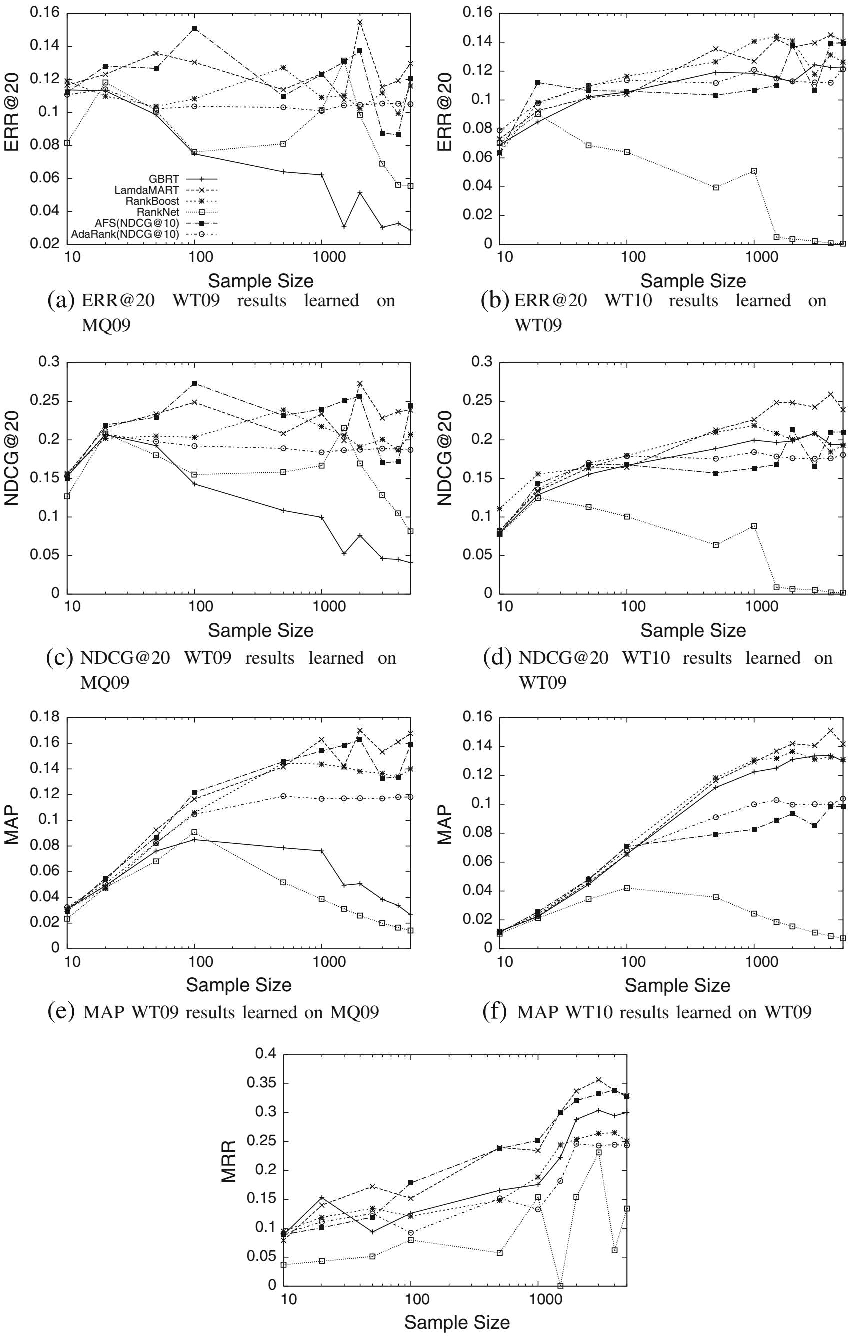
(d) NDCG@20 WT10 results learned on WT09

(e) MAP WT09 results learned on MQ09

(f) MAP WT10 results learned on WT09

(g) MRR NAV06 results

**Fig. 6** Effect on test effectiveness on the CW09B query sets of different sample sizes, obtained using the document representation that does not include anchor text, learned using various learning to rank techniques. Key is common to **a**–**g**



(a) ERR@20 WT09 results learned on MQ09

(b) ERR@20 WT10 results learned on WT09

(c) NDCG@20 WT09 results learned on MQ09

(d) NDCG@20 WT10 results learned on WT09

(e) MAP WT09 results learned on MQ09

(f) MAP WT10 results learned on WT09

(g) MRR NAV06 results

**Fig. 7** Effect on test effectiveness on the CW09B query sets of different sample sizes, obtained using the document representation that includes anchor text, learned using various learning to rank techniques. Key is common to **a**–**g**

achieved for samples smaller than 100 (WT09) or 1,000 (WT10). For MAP, 1,000–3,000 documents is necessary for maximal effectiveness. These results show that the choice of test measure is important when making a decision on sample size. Indeed, the ERR measure discounts the contributions of documents that appear after highly relevant documents, hence relevant documents retrieved further down the ranking matter less for ERR than, for example, NDCG. This explains why smaller sample sizes are more effective for ERR than NDCG. In contrast, MAP needs the largest sized samples, explained by its recall component.

For the purely navigational NAV06 query set, the trend is markedly different from the WT09 and WT10 sets, whereby large improvements are obtained for very large sample sizes. In particular, when anchor text is not used (Fig. 6g), a sample size of 5,000 markedly improves over smaller sample sizes, as many of the relevant documents are not found in these smaller samples. When the anchor text document representation is used for sampling (Fig. 7g), more relevant documents are found in the smaller samples, resulting in increased effectiveness for smaller samples, due to the ability of anchor text to identify homepages (Hawking et al. 2004). However, the trends for NAV06 do not mirror the HP04 and NP04 samples on the GOV corpus, in that a sample size of 50 is still insufficient for fully effective retrieval, even with the inclusion of anchor text in the document representation used for generating the sample.

The need for bigger samples for CW09B than GOV suggests that relevant documents occur at deeper ranks in the sample than for GOV, and hence the sample must be larger to compensate, even for navigational information needs. Indeed, we note that CW09B is approximately 50 times larger than the GOV corpus. It also appears to represent a more difficult corpus, i.e. identifying relevant documents is comparatively more challenging. For instance, the presence of spam documents in CW09B—as discussed by Cormack et al. (2011)—will likely cause relevant documents to be ranked after spam in the sample. On the other hand, in Fig. 7g, the benefit of extending past size 2,000 is less pronounced, suggesting that the sample size of 5,000 used by Craswell et al. (2010) is perhaps unnecessary for information needs that are purely navigational in nature.

### *5.1.2 Learning to rank technique analysis*

Taking each learning to rank technique in turn, we can make observations about their suitability for different sample sizes. To aid in this analysis, Table 7 shows the mean and standard deviation of the effectiveness of each learning to rank technique, for each query set and test evaluation measure.

In general, all techniques behave similarly with the exception of RankNet. Indeed, RankNet (denoted by (┈⊡┈) in Figs. 5, 6 and 7) generally does not perform as high as other techniques for the GOV query sets, while for WT09 and WT10, it degrades retrieval performance for all sample sizes larger than 20 when evaluated using NDCG@20 or ERR@20 measures, and 100 for MAP. This suggests that it is unable to resolve the additional pairwise constraints that are exponentially added as sample size increases. For the navigational query sets with less relevant documents, the number of pair constraints are less, and hence RankNet exhibits less sensitivity. In contrast to RankNet, RankBoost (┈⋇┈)—also a pairwise technique—performs similarly to other learning to rank techniques, showing somewhat more robustness to larger sample sizes. Nevertheless, from Figs. 5, 6 and 7, the performances of both RankBoost and RankNet are relatively better for smaller sample sizes on the CW09B query sets than for the GOV query sets. This is likely

**Table 7** For each query set and test evaluation measure, the mean performance of each learning to rank technique, across all sample sizes

| Query set | Test evaluation measure | Technique | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | LambdaMART | | GBRT | | RankNet | | RankBoost | | AFS | | AdaRank | |
| | | Mean | $\sigma$ | Mean | $\sigma$ | Mean | $\sigma$ | Mean | $\sigma$ | Mean | $\sigma$ | Mean | $\sigma$ |
| HP04 | MRR | .671 | .022 | .491 | .094 | .241 | .128 | .641 | .031 | .701 | .022 | **.718** | .028 |
| NP04 | MRR | .571 | .036 | .441 | .070 | .371 | .117 | .521 | .031 | .521 | .040 | **.659** | .031 |
| TD04 | MAP | **.198** | .045 | .181 | .044 | .051 | .028 | .171 | .040 | .161 | .033 | .181 | .037 |
| WT09 | ERR@20 | **.147** | .014 | .121 | .009 | .091 | .024 | .141 | .013 | .131 | .002 | .131 | .008 |
| WT09 | NDCG@20 | **.272** | .052 | .231 | .036 | .151 | .049 | **.272** | .052 | .251 | .019 | .261 | .027 |
| WT09 | MAP | .141 | .069 | .121 | .054 | .041 | .030 | **.153** | .071 | .141 | .050 | .151 | .053 |
| WT10 | ERR@20 | .101 | .016 | .101 | .016 | .051 | .045 | .111 | .017 | **.119** | .018 | .101 | .014 |
| WT10 | NDCG@20 | .161 | .040 | .161 | .042 | .071 | .065 | .171 | .044 | **.202** | .045 | .171 | .033 |
| WT10 | MAP | .081 | .044 | .081 | .047 | .021 | .014 | .081 | .044 | **.098** | .044 | .091 | .038 |
| NAV06 | MRR | **.153** | .091 | .141 | .098 | .071 | .057 | .121 | .066 | .141 | .081 | .121 | .057 |
| WT09$_a$ | ERR@20 | **.121** | .016 | .061 | .031 | .081 | .022 | .101 | .010 | .111 | .019 | .101 | .004 |
| WT09$_a$ | NDCG@20 | .211 | .044 | .101 | .055 | .141 | .041 | .191 | .034 | **.222** | .038 | .181 | .013 |
| WT09$_a$ | MAP | .111 | .053 | .051 | .022 | .031 | .023 | .101 | .046 | **.122** | .043 | .091 | .029 |
| WT10$_a$ | ERR@20 | **.116** | .029 | .101 | .022 | .031 | .031 | .111 | .028 | .111 | .021 | .111 | .011 |
| WT10$_a$ | NDCG@20 | **.189** | .067 | .161 | .050 | .051 | .045 | .181 | .063 | .161 | .036 | .161 | .029 |
| WT10$_a$ | MAP | **.092** | .055 | .081 | .050 | .021 | .012 | .081 | .051 | .071 | .028 | .071 | .033 |
| NAV06$_a$ | MRR | .231 | .098 | .181 | .087 | .081 | .064 | .171 | .072 | **.236** | .094 | .161 | .062 |
| Means | | **.211** | .047 | .161 | .049 | .091 | .047 | .191 | .042 | .201 | .037 | .201 | .030 |

The highest performing learning to rank technique in each row is highlighted. The last row shows the average of all numbers in each column due to the use of graded relevance assessments, which allow more pairwise preferences to be better expressed than for binary relevance assessments with the same size of sample.

GBRT (—+—) produces a robust retrieval performance, giving a reasonable effectiveness at small sample sizes. However, for WT09, its performance is not comparable to other effective techniques for larger sample sizes. On the other hand, the state-of-the-art LambdaMART (·····✳·····) exhibits effective performances across all sample sizes and query sets.

Lastly, the AFS and AdaRank listwise techniques also produce robust and effective overall performances. On the GOV query sets, AdaRank (--⊖--) is generally more effective. On HP04 and TD04, the difference is not marked, but it is more marked for NP04. However, for WT09, WT10 and NAV06, AdaRank is less effective for very large samples (≥1,000), while AFS is more stable.

To conclude, while RankNet exhibits the worst overall performance in Table 7, LambdaMART followed by AFS and AdaRank are the most effective techniques across all sample sizes. These performances are in line with LambdaMART's top performance in the Yahoo! Learning to Rank Challenge (Chapelle and Chang 2011), and the observations concerning the effectiveness of listwise techniques reported by Liu (2009).

### *5.1.3 Hypotheses analysis*

To address each of the hypotheses and research questions defined in Sect. 2.1.4, we deploy ANOVAs. In particular, Table 8 presents the *p* values of the within-subject ANOVAs

**Table 8** ANOVA *p* values for different query sets and evaluation measures as sample sizes or learners are varied

| Query set | Test evaluation measure | *p* value | | |
|---|---|---|---|---|
| | | Learning technique<br>– | Sample size<br>Hypothesis 1 | Technique × sample size<br>Hypothesis 3 |
| HP04 | MRR | 0.00508** | 0.58472 | 4.661e−12** |
| NP04 | MRR | 0.33373 | 0.03319* | 3.618e−15** |
| TD04 | MAP | 0.60300 | 8.101e−11** | 0.07133 |
| WT09 | ERR@20 | 0.8641 | 0.4333 | 0.5598 |
| WT09 | NDCG@20 | 0.9097 | 0.7989 | 0.5502 |
| WT09 | MAP | 0.6594 | 2.733e−06** | 0.2220 |
| WT10 | ERR@20 | 0.7825 | 0.1165 | 0.4441 |
| WT10 | NDCG@20 | 0.8310 | 0.1164 | 0.8359 |
| WT10 | MAP | 0.7101 | 2.275e−06** | 0.4241 |
| NAV06 | MRR | 0.9282 | 4.429e−05** | 0.6942 |
| WT09$_a$ | ERR@20 | 0.13107 | 0.03584* | 1.187e−05** |
| WT09$_a$ | NDCG@20 | 0.1400 | 0.3281 | 1.784e−06** |
| WT09$_a$ | MAP | 0.0909035 | 0.0009944** | 0.0002281** |
| WT10$_a$ | ERR@20 | 0.9773 | 0.2160 | 0.8562 |
| WT10$_a$ | NDCG@20 | 0.94809 | 0.07734 | 0.52887 |
| WT10$_a$ | MAP | 0.2986 | 1.598e−06** | 0.0690 |
| NAV06$_a$ | MRR | 0.8600 | 1.797e−06** | 0.8043 |

* Denotes a *p* value of less than 5 %, ** denotes a *p* value less than 1 %

computed over all queries within a given query set, while varying the sample size and learning to rank technique. The *a* suffix denotes when the sample for a query set was obtained using anchor text. For example, WT09$_a$ MAP represents the ANOVA calculated across all WT09 queries, learning to rank techniques, and sample sizes, where the sample document representation includes anchor text, and the queries are evaluated using the MAP test evaluation measure. From the ANOVAs, we exclude RankNet, for the reasons stated in Sect. 5.1.2, as we found that its high variance under large samples introduced sensitivities to sample size not present for the other learning to rank techniques.

In Hypothesis 1, we postulated that sample size should affect retrieval performance. On consideration of the Sample Size column of Table 8, we find that sample size significantly impacts on the NP04, TD04 and NAV06 query sets, as well as WT09 and WT10 for only the MAP test evaluation measure (a single outlier is WT09$_a$ evaluated by ERR@20, which is explained below). Indeed, for MAP, deeper samples are required than for other measures. In general, the results in Table 8 assert the importance of the sample size for effective learning, and mirror our general observations across Figs. 5, 6 and 7. Overall, we conclude that Hypothesis 1 is generally validated.

Hypothesis 2 is concerned with the types of information needs. In particular, we argued that the impact of the sample size will vary for different information needs, while also postulating that the use of anchor text in the sampling document representation could affect the results. From Fig. 5a, we observed that for the navigational HP04 query set, a deep sample is not required for effective results. This is mirrored in Table 8, where we find that sample size has no significant impact for HP04 effectiveness. This contrasts with, for example, the informational TD04 query set, where a significant dependence ($p = 8.101e{-}11$) is observed, as would be expected from Fig. 5c. However, for the NAV06 query set, a significant dependence on sample size is observed, regardless of whether anchor text is present in the sampling document representation. Indeed, anchor text markedly improves the effectiveness of smaller samples for navigational queries—e.g. while in Fig. 5a the trend for HP04 is nearly flat, NAV06 (Fig. 7g) has a markedly improved effectiveness for sample sizes 100–4,000 when anchor text is used. However, anchor text for NAV06 does not exhibit the flat trend observed for HP04, suggesting that large samples are still necessary on the larger ClueWeb09 corpus, as illustrated by the significant *p* values for NAV06 and NAV06*a* in the Sample Size column of Table 8 ($p = 4.429e{-}05$ and $1.797e{-}06$, respectively). To illustrate this, Table 9 reports the recall measure for samples with sizes 1,000 and 5,000, with and without anchor text. For the entirely navigational NAV06 query set, more relevant documents are ranked in the top 1,000 documents of the sample when using anchor text, explaining the improved effectiveness.

On the other hand, for the mixed queries of WT09 and WT10, comparing across Figs. 6 and 7, the general magnitude of effectiveness values are mostly unchanged by the addition of anchor text. We note a greater variation between the performances of different learning

**Table 9** Recall of samples of size 1,000 and 5,000 for the ClueWeb09 query sets, as the presence of anchor text in the sample document representation is varied

| Query set ↓ | 1,000 | | 5,000 | |
|---|---|---|---|---|
| Anchor text → | × | ✔ | × | ✔ |
| WT09 (%) | 58.2 | 55.2 | 78.6 | 75.9 |
| WT10 (%) | 31.6 | 30.2 | 38.9 | 37.5 |
| NAV06 (%) | 20 | 34 | 56 | 48 |

to rank techniques on WT09 when anchor text is deployed, suggesting that for these queries, anchor text produces a noisier sample, in which some learning to rank techniques struggle to properly rank relevant documents. This is manifested by significant values for $\mathrm{WT09}_a$ in Table 8 (e.g. $p = 0.03584$). This may be due to the contrasting nature of the WT09 query set and its corresponding MQ09 training queries. However, in general, from Table 9, we see that sampling with anchor text reduces the recall for the mixed Web track query sets. The likely cause of this is that anchor text is often spammed to improperly manipulate Web search results (Castillo et al. 2006), which may explain the adverse effect of the anchor text samples on recall. On the other hand, for the GOV corpus that only contains documents and links administered by various US government departments, there are less adversarial issues relating to anchor text.

In summary, we find that Hypothesis 2 is partially validated: the impact of the sample size can depend on the type of information need, while the presence of anchor text is important for assuring the effectiveness of smaller sample sizes for navigational queries.

Hypothesis 3 stipulates that the effectiveness of learned models depends on both the choice of the learning to rank technique and the sample size. To analyse this hypothesis, we use the last column of Table 8, which denotes significant dependencies on both of these independent variables. In general, we observe significant dependencies between effectiveness and the learning to rank technique only for the HP04 and NP04 query sets, as well as $\mathrm{WT09}_a$. On inspection of the corresponding figures for these query sets (namely Figs. 5a, b, 7a, c, e), this observation can be explained as follows: For these query sets and corresponding test evaluation measures, there are learning to rank techniques that markedly degrade in performance for large sample sizes (e.g. GBRT, in addition to the excluded RankNet), thereby diverging from the other effective techniques. Overall, we find Hypothesis 3 to be partially validated, as GBRT, and in fact RankNet, markedly degrade in effectiveness for large sample sizes.

For completeness, Table 8 also reports the ANOVA $p$ values for the dependence of effectiveness on the selection of learning to rank technique alone. In the results shown in the Learning Technique column of Table 8, a significant dependence is only exhibited for the HP04 query set. Indeed, on inspection of the corresponding Fig. 5a for HP04, we observe a concordance in the relative ordering of the learning to rank techniques across the sample sizes. Hence, the choice of the learning to rank technique has a significant impact on retrieval effectiveness for this query set. For all other query sets, there is no significant dependence between the selected learning to rank technique and effectiveness, showing that many learning to rank techniques have very similar performances.

Finally, Research Question 1 is concerned with identifying what aspects may have an impact on the smallest effective sample size. To address this research question, Table 10 reports the minimum sample size for each learning to rank technique, query set and test measure that does not result in a significantly degraded performance compared to the sample size that is most effective. On analysing this table, we note that the mean minimum sample size varies across both query sets and test evaluation measures. For instance, the informational TD04 query set needs a large sample to ensure effectiveness is not significantly degraded from the best achieved, while the navigational HP04 exhibits effective results across all learning to rank techniques with a sample of only 10 documents. Evaluating the Web track query sets using NDCG@20 and ERR@20 measures permit small effective samples (with ERR@20 needing smaller samples than NDCG@20), while the smallest effective sample for MAP is markedly larger, due to its emphasis on recall. For the NAV06 query set, the single relevant document for each query necessitates deep samples, but sample size could be reduced by using anchor text within the sampling document

**Table 10** Smallest sample size not statistically degraded from the most effective sample size for a given query set, test evaluation measure and learning to rank technique

| Query set | Measure | Mean | Technique | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | GBRT | LambdaMART | RankBoost | RankNet | AFS | AdaRank |
| HP04 | MRR | 10.0 | 10 | 10 | 10 | 10 | 10 | 10 |
| NP04 | MRR | 65.0 | 200 | 20 | 50 | 50 | 20 | 50 |
| TD04 | MAP | 425.0 | 400 | 800 | 700 | 50 | 100 | 500 |
| WT09 | ERR@20 | 13.3 | 10 | 20 | 10 | 10 | 10 | 20 |
| WT09 | NDCG@20 | 25.0 | 20 | 20 | 20 | 20 | 20 | 50 |
| WT09 | MAP | 500 | 2,175.0 | 2,000 | 1,500 | 50 | 5,000 | 4,000 |
| WT10 | ERR@20 | 20.0 | 10 | 20 | 10 | 10 | 50 | 20 |
| WT10 | NDCG@20 | 543.3 | 50 | 2,000 | 100 | 10 | 1,000 | 100 |
| WT10 | MAP | 1,933.3 | 1,500 | 4,000 | 1,000 | 100 | 4,000 | 1,000 |
| NAV06 | MRR | 5,000 | 5,000 | 5,000 | 5,000 | 5,000 | 5,000 | 5,000 |
| WT09$_a$ | ERR@20 | 20.0 | 10 | 50 | 10 | 20 | 20 | 10 |
| WT09$_a$ | NDCG@20 | 25.0 | 20 | 50 | 20 | 20 | 20 | 20 |
| WT09$_a$ | MAP | 525.0 | 50 | 1,000 | 500 | 100 | 1,000 | 500 |
| WT10$_a$ | ERR@20 | 188.3 | 50 | 500 | 500 | 10 | 20 | 50 |
| WT10$_a$ | NDCG@20 | 678.3 | 500 | 1,000 | 500 | 20 | 2,000 | 50 |
| WT0$_a$ | MAP | 1,591.7 | 1,500 | 4,000 | 1,000 | 50 | 1,500 | 1,500 |
| NAV06$_a$ | MRR | 1,083.3 | 1,500 | 1,500 | 1,000 | 1,000 | 1,000 | 500 |
| Means | | 842.5 | 666.5 | 1,293.5 | 701.8 | 384.1 | 1,221.8 | 787.1 |

Significant differences are measured using the paired $t$ test, with $p < 0.01$. The $a$ subscript for ClueWeb09 query sets denotes when anchor text is used within the sampling document representation

representation (see Table 10, NAV06 and NAV06$_a$ rows). However, for the other ClueWeb09 query sets, namely WT09 and WT10, adding anchor text did not markedly change the smallest effective sample size. Lastly, some learning to rank techniques were less sensitive than others: for instance, RankNet prefers a small sample, but this is mainly due to its ineffective performance at large sample sizes; AFS and the state-of-the-art LambdaMART require similarly large samples (mean approx. 1,200), while AdaRank and GBRT were effective with smaller samples (mean approx. 600–700). Overall, we conclude that while the smallest effective sample size can depend on a number of factors (sampling document representation, type of information need), using test evaluation measures that focus on the top-ranked documents (e.g. ERR@20 rather than MAP) can reduce the size of the minimum effective sample. Moreover, using anchor text when sampling for navigational information needs can reduce the minimum size of an effective sample without a marked impact on other information need types.

### *5.1.4 Summary*

Our detailed experiments have permitted various conclusions to be drawn with respect to the sample size. In the following, we list the hypotheses and research question from Sect. 2.1.4, and summarise our findings for each.

**Hypothesis 1** The observed effectiveness of learned models can be affected by different sample sizes.

Generally Validated: Sample size has a significant impact on all query sets except the navigational HP04. The choice of test evaluation measure can affect the impact of sample size for instance, for the WT09 and WT10 query sets evaluated by NDCG@20, sample size did not have significant impact, while for MAP it did.

**Hypothesis 2** The observed effectiveness of learned models can be affected by the type of information need observed in the queries, and the used document representation for generating the samples, regardless of the size of these samples.

Partially validated: Adding anchor text to the NAV06 query set improved retrieval performance for smaller sample sizes. For the mixed WT09 and WT10 query sets, sample size was not significantly important.

**Hypothesis 3** The observed effectiveness of learned models depends on the deployed learning to rank technique and the sample size.

Partially validated: The effectiveness of the learned models depends on the sample size for some learning to rank techniques, namely GBRT and RankNet.

**Research question 1** *What are the aspects that define the smallest sample size for an effective learned model?*

The test evaluation measure, the type of information need and the learning to rank technique have all been shown to have an impact on the size of the smallest effective sample. In particular, from Table 10, we found that the smallest effective sample size was generally 10–50 documents for the navigational query sets on the GOV (LETOR) test collection, while 400 documents were necessary for the informational TD04 query set. For the TREC Web track query sets of mixed information needs on the much larger ClueWeb09 corpus, 20–50 documents were sufficient for effective ERR@20 performances, while larger samples were required for some techniques and query sets to ensure effective NDCG@20 (e.g. LambdaMART: 20 documents for WT09 vs. 2,000 documents for WT10). Furthermore, for an effective MAP performance, samples of 2,000 documents were necessary across all learning to rank techniques. For the navigational query set on ClueWeb09, we found that it was important to use anchor text in the sampling document representation to ensure effective retrieval at sample sizes smaller than 5,000 documents.

In summary, to ensure effective retrieval across different types of information needs for the ClueWeb09 corpus, our results suggest that a sample for learning to rank is created using at least 1,500 documents obtained using a document representation that includes both the body of the document and the anchor text of incoming hyperlinks, or at least 2,000 documents when not using anchor text, to ensure maximum effectiveness.

## 5.2 Learning evaluation measure & cutoff

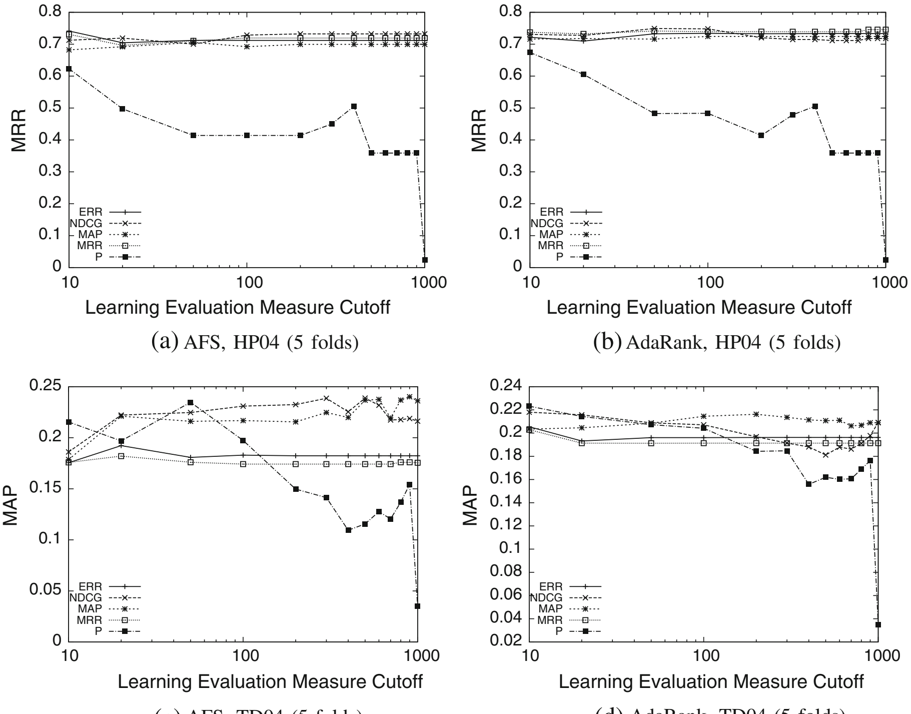
For listwise learning to rank techniques, we postulated in Sect. 2.2.3 that the choice of the learning evaluation measure and rank cutoff used within their loss function may have an impact on the effectiveness of the resulting learned model. This section addresses this second research theme, by identifying appropriate measures and cutoffs for listwise learning to rank techniques. For these experiments, as described in Sect. 3, we fix the sample size. In particular, following the results of Sect. 5.1, we choose the maximal sample sizes (1,000 for GOV, 5,000 for CW09B), as these have the highest potential to perform well across all query sets, while the sample document representation for CW09B is fixed to exclude anchor text (again, this is also an effective setting across all query sets). In the following, we present and analyse our experiments, firstly, in a graphical manner (Sect. 5.2.1), and secondly in an empirical manner (Sect. 5.2.2). We then summarise whether our hypotheses from Sect. 2.2.3 were validated (Sect. 5.2.3).

### 5.2.1 Graphical analysis

Figure 8 (GOV query sets) and Fig. 9 (CW09B query sets) show the impact of using different learning evaluation measures with different rank cutoffs for AFS and AdaRank on the effectiveness for the various query sets. For reasons of brevity, we omit figures for the NP04 and NAV06 query sets, as the trends observed are identical to HP04 (Fig. 8a, b)

Firstly, on analysis of the navigational HP04 query set (Fig. 8a, b), is it clear that the learning evaluation measure and its rank cutoff do not impact greatly the effectiveness of the learned models, with the exception of the precision measure (P). Indeed, as expected, the low informativeness of the precision (P) measure makes it unsuitable for use as a learning evaluation measure at all but the smallest depths. For the other measures, the variability is much lower across different cutoff depths, and the measures have widely similar performances. This is due to the very few relevant document for each query in this set (see Table 2), which ensures that the various measures can only respond to changes where the relevant document moves up or down in the ranking.

For the more informational TD04 query set (Fig. 8c, d), there is more contrast between the performance of the various learning evaluation measures than for HP04. In particular,



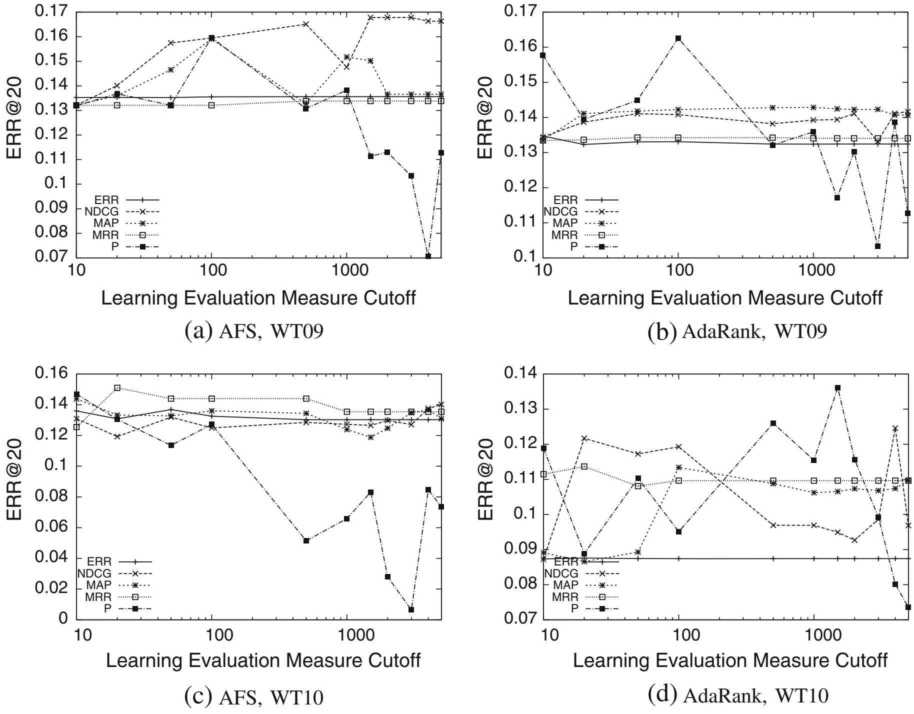
(a) AFS, HP04 (5 folds)

(b) AdaRank, HP04 (5 folds)

(c) AFS, TD04 (5 folds)

(d) AdaRank, TD04 (5 folds)

**Fig. 8** Effect on test measure effectiveness on the HP (in terms of MRR) and TD (MAP) GOV query sets of different learning evaluation measures and rank cutoffs, using AFS or AdaRank. Sample size 1,000. NP figures are omitted, as these are very similar to HP



(a) AFS, WT09

(b) AdaRank, WT09

(c) AFS, WT10

(d) AdaRank, WT10

**Fig. 9** Effect of different learning evaluation measures and rank cutoffs on test ERR@20 on the WT09 and WT10 CW09B query sets (NAV06 shows a similar pattern to the other HP04 and NP04 navigational query sets), learned using AFS and AdaRank. Sample size 5,000

for both AFS and AdaRank, MRR and the related ERR measures are very similar in performance, and their rank cutoff has little impact on the effectiveness of the learned model. In contrast, learned models obtained using the MAP and NDCG learning evaluation measures exhibit generally higher performance than ERR and MRR, due to their higher informativeness.

On analysing WT09 and WT10 query sets in Fig. 9, we observe a higher variance in effectiveness for some learning evaluation measures across different sample sizes for these query sets than for TD04. In particular, some learned models using precision can be effective, but such occurrences are fairly seldom and appear to occur randomly. Once again, MAP and NDCG appear to be effective learning evaluation measures, with the effectiveness of NDCG increasing as rank cutoff increases for WT09 using AFS (Fig. 9a). In contrast, with their focus on the first-ranked relevant documents, the ERR and MRR learning evaluation measures show little variance in effectiveness as rank cutoff increases. However, we note that the WT10 learned models are more markedly effective for MRR than ERR—an observation not found for the other query sets.

### 5.2.2 *Empirical analysis*

To aid in the cross-comparison of measures, for both AFS and AdaRank, Table 11 shows the mean and standard deviation of the test performance measure across all cutoffs for a given learning evaluation measure. For instance, the lines for ERR (—+—) in Figs. 8 and 9

**Table 11** For AFS and AdaRank, mean and standard deviation of the test performance measure of learned models obtained using different learning evaluation measures across all rank cutoffs

| Query set | Test measure | Learning evaluation measure | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ERR | | NDCG | | MAP | | MRR | | P | |
| | | Mean | $\sigma$ | Mean | $\sigma$ | Mean | $\sigma$ | Mean | $\sigma$ | Mean | $\sigma$ |
| *AFS* | | | | | | | | | | | |
| HP04 | MRR | .713 | .008 | **.724** | .013 | .697 | .005 | .717 | .008 | .412 | .140 |
| NP04 | MRR | .405 | .042 | **.423** | .035 | .412 | .037 | .415 | .037 | .220 | .154 |
| TD04 | MAP | .182 | .004 | **.220** | .017 | .220 | .019 | .175 | .002 | .151 | .050 |
| WT09 | ERR@20 | .136 | .000 | **.158** | .012 | .141 | .009 | .133 | .001 | .122 | .022 |
| WT09 | NDCG@20 | .271 | .002 | **.309** | .022 | .285 | .016 | .263 | .001 | .242 | .049 |
| WT09 | MAP | .188 | .002 | **.205** | .012 | .193 | .010 | .176 | .001 | .168 | .035 |
| WT10 | ERR@20 | .132 | .002 | .129 | .006 | .132 | .007 | **.138** | .007 | .083 | .042 |
| WT10 | NDCG@20 | .198 | .002 | .234 | .013 | **.248** | .010 | .209 | .005 | .141 | .070 |
| WT10 | MAP | .097 | .000 | .135 | .003 | **.137** | .002 | .098 | .001 | .096 | .027 |
| NAV06 | MRR | **.422** | .030 | **.422** | .030 | **.422** | .030 | **.422** | .030 | .161 | .087 |
| *AdaRank* | | | | | | | | | | | |
| HP04 | MRR | .730 | .007 | .724 | .014 | .723 | .003 | **.740** | .004 | .439 | .160 |
| NP04 | MRR | .669 | .002 | .673 | .013 | .670 | .002 | **.679** | .001 | .308 | .194 |
| TD04 | MAP | .196 | .003 | .198 | .011 | **.209** | .004 | .193 | .004 | .174 | .044 |
| WT09 | ERR@20 | .133 | .001 | .139 | .003 | **.141** | .003 | .134 | .000 | .134 | .017 |
| WT09 | NDCG@20 | .260 | .001 | .272 | .004 | **.274** | .003 | .261 | .000 | .270 | .035 |
| WT09 | MAP | .184 | .001 | .190 | .002 | **.192** | .002 | .184 | .000 | .185 | .021 |
| WT10 | ERR@20 | .087 | .000 | .104 | .013 | .103 | .009 | **.110** | .001 | .105 | .019 |
| WT10 | NDCG@20 | .148 | .000 | .172 | .014 | **.175** | .012 | .162 | .001 | .164 | .027 |
| WT10 | MAP | .097 | .000 | .113 | .005 | **.116** | .007 | .089 | .001 | .105 | .012 |
| NAV06 | MRR | **.260** | .000 | .258 | .003 | .241 | .012 | .241 | .012 | .171 | .094 |
| *Means* | | .275 | .005 | **.290** | .012 | .287 | .010 | .277 | .006 | .193 | .065 |

The highest performing learning to rank technique in each row is highlighted. The last row shows the average of all numbers in each column

are summarised in the ERR column of Table 11. The last row in Table 11 reports the average in each column, to indicate the trends across the different learning evaluation measures.

Comparing the learning evaluation measures, we observe from Table 11—similar to the performances observed in Figs. 8 and 9 —that Precision (P) provides the lowest performance, and the highest variance, as the learned models it produces are not always effective (the high variance is explained in that some models consist of only a single effective feature, while in other scenarios a low quality model is obtained). Of the other measures, NDCG and MAP are overall the most effective learning evaluation measures, followed by MRR and ERR in certain cases.

Breaking this down by query set, for the HP04 and NP04 navigational sets, we observe from Table 11 that NDCG is the most effective learning evaluation measure for AFS, followed by MRR (mean performances of 0.724 and 0.717, respectively, for HP04, 0.423 and 0.415 for NP04). While it is expected that the informative NDCG measure produces the most effective models, the promising performances of MRR and the related ERR measure suggest that, for these query sets, learning using just the top of the ranking is sufficient to obtain effective learned models. This is also true for AdaRank learned models, with MRR exhibiting the most effective learned models for both query sets (0.740 and 0.679).

For the informational TD04 query set, MAP and NDCG are effective choices, with similar results observed for the WT09 and WT10 mixed query sets, regardless of the test evaluation measure. Indeed, for WT09 and WT10, it is not the case that learning using the test evaluation measure results in the most effective models when evaluated using the test measure. This is particularly true for learning using ERR, which always results in lower quality models than NDCG, even when evaluated by ERR@20 (for instance, for AFS, models learned using NDCG have mean ERR@20 of 0.158, versus 0.136 for ERR as a learning evaluation measure). Moreover, this agrees with the theoretical speculations of Robertson (2008), and validates the results of Yilmaz and Robertson (2010) across many more query sets and in a reproducible setting. The observed results also confirm the earlier work of He et al. (2008), but which was not conducted within a learning to rank setting.

Comparing the AdaRank and AFS listwise learning to rank techniques, from Table 11, we observe that the mean performances for both techniques are broadly similar-in line with Table 7. However, by using a cell-by-cell comparison, we find AdaRank to perform slightly better than AFS, while also exhibiting an overall lower variance.

Overall, we observe the following ranking of learning evaluation measures: NDCG $\geq$ MAP $\geq \{ERR, MRR\} \gg$ P, and hence recommend NDCG as the most effective learning measure. For making an appropriate choice of measure, high effectiveness should be regarded as more important than low variance. Indeed, while ERR has a low variance, it performs lower than MAP or NDCG. Even for the effective MAP and NDCG measures, the variance was low suggesting that to permit effective learning, evaluating using a small cutoff (e.g. 10) is sufficient, due to the top-heavy nature of most evaluation measures. Moreover, from Fig. 2, we note that smaller rank cutoffs provide efficiency advantages by markedly reducing learning time.

Finally, we address our hypotheses concerning the evaluation measure choice (Hypothesis 4) and rank cutoff (Hypothesis 5). Similar to Sect. 5.1 above, this is achieved with the aid of ANOVAs, which are reported in Table 12. By analysing the Learning Evaluation Measure column of Table 12, we observe significant dependencies on the selected measure for AFS and AdaRank for 6 and 5 out of 9 query sets, respectively. This suggests that the choice of measure can have an impact on effectiveness, particularly for the GOV query sets, partially upholding Hypothesis 4.

This observation about the GOV query sets is also mirrored in the results of the Learning Evaluation Measure Cutoff column of Table 12, with AdaRank showing significant dependence on the rank cutoff. However, the results for the CW09B query sets do not exhibit significant dependencies on evaluation measure cutoff, which is explained by the higher variance and lack of concordance of learning to rank techniques across different cutoffs for these query sets. Indeed, only from inspection of Figs. 8 and 9 do we note the dependence of effectiveness on the cutoff of the precision measure. Overall, we conclude that due to the focus of most evaluation measures on the top-ranked documents, only for the precision measure can Hypothesis 5 be validated.

### *5.2.3 Summary*

Our detailed experiments have permitted various conclusions to be drawn with respect to the choice of learning evaluation measure rank cutoff. Below, we summarise our findings for the hypotheses we defined in Sect. 2.2.3:

**Table 12** ANOVA *p* values for different query sets and evaluation measures as learning evaluation measure and cutoffs are varied

| Query set | Test measure | *p* value | | |
|---|---|---|---|---|
| | | Learning evaluation<br>Measure<br>Hypothesis 4 | Learning evaluation<br>Measure cutoff<br>Hypothesis 5 | Learning evaluation<br>Measure × cutoff<br>– |
| *AFS* | | | | |
| HP04 | MRR | 2e−16** | 0.06317 | 0.11763 |
| NP04 | MRR | 2.2e−16** | 0.2936 | 4.936e−07** |
| TD04 | MAP | 5.213e−09** | 0.340027 | 0.002777** |
| WT09 | ERR@20 | 0.1266 | 0.5082 | 0.5630 |
| WT09 | NDCG@20 | 0.2587 | 0.6254 | 0.7079 |
| WT09 | MAP | 0.5921 | 0.7468 | 0.8490 |
| WT10 | ERR@20 | 6.408e−09** | 0.1529 | 0.2954 |
| WT10 | NDCG@20 | 0.001968** | 0.281312 | 0.413536 |
| WT10 | MAP | 0.2141 | 0.3738 | 0.9089 |
| NAV06 | MRR | 6.098e−07** | 0.3730 | 0.8111 |
| *AdaRank* | | | | |
| HP04 | MRR | 2.2e−16** | 0.0194766* | 0.0006706** |
| NP04 | MRR | 2.2e−16** | 0.0004233** | 2.522e−09** |
| TD04 | MAP | 0.0007071** | 0.0265877* | 0.0183190* |
| WT09 | ERR@20 | 0.5336 | 0.9631 | 0.9986 |
| WT09 | NDCG@20 | 0.9796 | 0.9392 | 0.9969 |
| WT09 | MAP | 0.9956 | 0.7389 | 0.9899 |
| WT10 | ERR@20 | 0.001188** | 0.543233 | 0.352908 |
| WT10 | NDCG@20 | 0.4031 | 0.8359 | 0.7595 |
| WT10 | MAP | 0.03274* | 0.62433 | 0.91968 |
| NAV06 | MRR | 0.4288 | 0.6982 | 0.9766 |

**Hypothesis 4** The observed effectiveness of the learned model obtained from a listwise learning to rank technique can be affected by the choice of the learning evaluation measure.

Partially validated: The choice of learning evaluation measure can significantly impact the effectiveness of learned models from listwise learning to rank techniques, but only for some types of information needs. In particular, for navigational queries with very few relevant documents, all measures except precision are comparably effective. However, for query sets with informational needs, measures such as MAP and NDCG are the most effective. Overall, regardless of the measure used to evaluate the effectiveness of the learned model, our results suggest that NDCG is an effective choice as a learning evaluation measure for a listwise learning to rank technique, due to its informative nature.

**Hypothesis 5** The observed effectiveness of the learned model obtained from a listwise learning to rank technique can be affected by the choice of the rank cutoff of the learning evaluation measure.

Partially validated: For the precision measure, there is a clear dependence between the learning evaluation measure rank cutoff and the effectiveness of the resulting learned models (see Figs. 8, 9). On the other hand, significant dependencies between rank cutoff and effectiveness are only observed for AdaRank on the GOV query sets, which can be considered as outliers. Therefore, given the reduced learning time achievable when using a smaller rank cutoff (see Fig. 2), our results suggest that 10 is a suitable rank cutoff for the learning evaluation measure of a listwise learning to rank technique.

## 5.3 Sample size and learning evaluation measure cutoff

In the preceding sections, we have examined the research themes concerning the sample size and the evaluation measure cutoff in independence, by fixing one while varying the other. However, in this section, we vary both concurrently to determine if there is dependence between these factors, thereby addressing our final research theme and its corresponding Hypothesis 6. This is performed by analysing the resulting performance of a learned model as the two factors are varied (Sect. 5.3.1), and by comparing the weights assigned to features in the learned models (Sect. 5.3.2). We summarise our observations in Sect. 5.3.3.

### *5.3.1 Graphical and empirical analysis*

For the WT10 query set (our most recent CW09B query set), we fix the learning to rank technique to AdaRank trained by NDCG, on a sample obtained using a document representation without anchor text. Indeed, based on the results in Sect. 5.1, AdaRank is an effective learning to rank technique, while based on Sect. 5.2, NDCG is the most effective learning measure, particularly on the CW09B query sets. Finally, the results in Sect. 5.1 show that anchor text has little impact on the results for WT10.

To measure the dependence between the sample size and the learning evaluation measure cutoff, we vary both as independent variables, and observe the resulting effectiveness as the dependent variable. Figure 10 presents the resulting NDCG@20 surface—the triangular shape is caused by the omission of points where the measure cutoff exceeds sample size. From this figure, we observe that effectiveness increases as sample size increases towards 1,000, but decreases past 1,000 (as expected from Fig. 6d for AdaRank). With respect to the measure cutoff, the surface is flat across all sample sizes. This suggests that there is no dependence between sample size and measure cutoff. Indeed, using a within-subject ANOVA, a $p$ value of 0.809 validates the lack of a significant dependence between effectiveness and the two independent variables.

### *5.3.2 Learned model comparison analysis*

Another method to determine the impact of the sample size and the learning measure cutoff is to compare and contrast the trained models. In particular, the AdaRank learning to rank technique calculates a weight $\alpha_f$ for each feature, where unselected features have $\alpha_f = 0$. From this, the vector $\boldsymbol{\alpha}$ contains the weights for all features. We normalise this vector to magnitude 1. Next, these weights are grouped by the five feature classes defined in Sect. 4.1 (weighting models, link analysis, etc.). Then by summing the absolute feature weights for all features with that class, we can determine the importance to the learned model, known as *mass*, of the features of that class:

$$mass(class) = \sum_{f \in features(class)} |\alpha_f| \tag{1}$$

where *features*(*class*) is the set of features belonging to that *class* from Sect. 4.1. Finally, we can compare models by measuring the difference in mass for each class of features between the two models.

We choose three points from the test performances for WT10 shown in Fig. 10:

- sample size of 10; learned using NDCG@10, denoted in Fig. 10 by ⊙;
- sample size of 1,000; learned using NDCG@10, denoted by ×;
- sample size of 1,000; learned using NDCG@1,000, denoted by +.

By comparing the corresponding learned models, we can examine the difference in feature weight mass between different sample sizes, and between different learning evaluation measure cutoffs. In particular, the left hand side of Fig. 11 shows the difference in feature weight mass between the learned models with sample size of 10 learned using NDCG@10 and sample size of 1,000 learned using NDCG@10. A positive difference denotes a feature group stronger in the former model. Similarly, the right hand side shows the difference in feature weight mass in models using sample size of 1,000 learned using NDCG@1,000 and sample size of 1,000 learned using NDCG@10.

From Fig. 11 (left), we observe that the link analysis features have markedly more emphasis in the models obtained for sample size 10. This is explained as follows: with a small sample size, the link analysis features can be applied more aggressively, as the

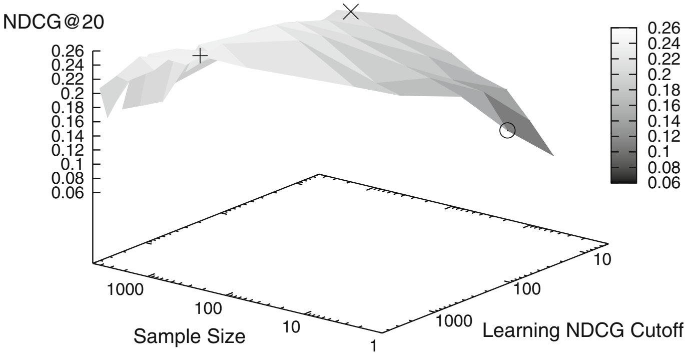


**Fig. 10** Surface plot for WT10 NDCG@20 performance, when learned using AdaRank(NDCG)

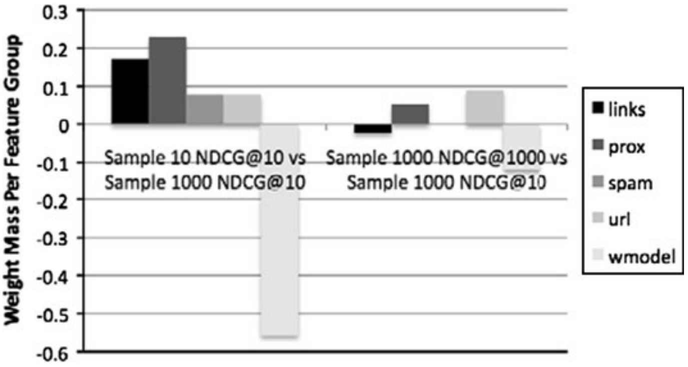


**Fig. 11** Relative feature mass for different feature classes and different learning settings

probability of relevance of the sample of documents being re-ranked is high (even if the effectiveness of the sample is lower). In contrast, for a larger sample, the probability of relevance of the documents in the sample is lower, and hence the weighting models are regarded as a stronger features. For example, consider a sample of two documents with two features: BM25 (used to identify the sample), and PageRank. For such a sample, it may be effective to re-rank the two documents entirely by PageRank, as it is likely that the second ranked document is relevant. However, there is much more potential to damage effectiveness by re-ranking a sample of 1,000 documents by PageRank, as the likelihood of relevance of the document with the highest PageRank (potentially at rank 1,000) is lower. This also explains why learned models obtained from a small sample should not be applied on larger samples and vice versa.

In Fig. 11 (right), we observe that for a large sample, the overall weights given to the feature groups are more similar, whether NDCG@10 or NDCG@1,000 is used for learning. However, for models obtained using NDCG@1,000, URL and proximity features play more of a role at getting relevant documents retrieved at the deeper ranks. The disparity between the most highly weighted feature classes from the different settings gives support to the fact that models emphasising different features are obtained according to the selected sample size and learning evaluation measure cutoff, even if the effectiveness of the models are similar.

### *5.3.3 Summary*

Through the above experiments and analysis, the dependence between the sample size and the learning evaluation measure has addressed our final hypothesis from Sect. 2.2.3:

**Hypothesis 6** The observed effectiveness of the learned model obtained from a listwise learning to rank technique can be affected by both the choice of the rank cutoff of the learning evaluation measure and the size of the samples.

Partially validated: We find that the choice of the learning evaluation measure cutoff for an effective model is not dependent on the sample size. Yet the models selected for different sample sizes and for different learning evaluation measures differ in the selected features. In particular, link analysis is a strong feature for ranking a small sample of 10 documents. However, for a large sample, when a learning evaluation measure with a large rank cutoff is used, link analysis features are introduced to improve the effectiveness of the ranking at lower ranks. Still, these features are not important for ordering the top-ranked documents. Overall, this suggests that using a larger sample does not require a larger rank cutoff for the learning evaluation measure used by the listwise learning to rank technique.

## 6 Conclusions

This study investigated how best to deploy learning to rank to obtain effective learned models, across different information need types, learning to rank techniques and corpora. It represents a larger and more thorough study than any currently present in the literature, investigating practical issues that arise when deploying learning to rank. In particular, we define three research themes, with corresponding hypotheses and research questions. In the first theme, we address the size and constitution of the sample for learning to rank (*when* to stop ranking). Moreover, in the second research theme, we address the choice of the learning evaluation measure and the corresponding rank cutoff for listwise learning to rank techniques (*how* to evaluate the learned models within the loss function of listwise learning to rank techniques). Lastly, our final research theme investigates the dependence between the sample size and the learning evaluation measure rank cutoff.

Overall, we found that *when* to stop ranking—the smallest effective sample—varied according to several factors: the information need, the evaluation measure used to test the models, and the presence of anchor text in the document representation used for sampling. In particular, from Table 10, we found that the smallest effective sample size was 10–50 documents for navigational information needs on the GOV (LETOR) test collection, while 400 documents were necessary for the topic distillation query set. For the TREC Web track query sets of mixed types of information needs on the much larger ClueWeb09 corpus, samples with as little as 20 documents are sufficient for effective ERR@20 performances. Some techniques and query sets were shown to require larger samples (up to 2,000 documents) for effective NDCG@20. Furthermore, for an effective MAP performance, samples of 2,000 documents were shown to be necessary for all learning to rank techniques. For the navigational query set on ClueWeb09, we found that it was important to use anchor text in the sampling document representation to ensure effective retrieval at sample sizes smaller than 5,000 documents—indeed, a sample size of at least 1,500 documents guarantees effective retrieval for all query sets using this representation. These results suggest that deep samples are necessary for effective retrieval in large Web corpora, indeed deeper than some recent learning to rank test collections such as those listed in Table 2. In addition, our experiments also showed that the effectiveness of learned models are generally dependent on the sample size (Hypothesis 1), partially dependent on the type of information need and the sample document representation (Hypothesis 2), and partially dependent on the choice of the learning to rank technique and the sample size (Hypothesis 3).

With respect to our second research theme addressing *how* the loss function for listwise learning to rank techniques should be defined—i.e. the choice of learning evaluation measures deployed by listwise learning to rank techniques—we found that the choice of the learning evaluation measure can indeed have an impact upon the effectiveness of the resulting learned model (Hypothesis 4), particularly for informational needs. Indeed, our results show that NDCG and MAP are the most effective learning evaluation measures, while the less informative ERR was not as effective, even when the test performance is evaluated by ERR. For the learning evaluation measure rank cutoff (Hypothesis 5), we only found the effectiveness of learned models to be markedly impacted by the rank cutoff for the precision measure.

Finally, for our third research theme, we showed that while there is no dependence between the learning measure cutoff and the sample size in terms of the effectiveness of the learned model (Hypothesis 6), the weights of the selected features can markedly differ between small and large samples, and between small and large learning evaluation measure cutoffs.

To summarise our empirical findings for applying learning to rank on a large Web corpus such as ClueWeb09, where evaluation is conducted using a measure such as NDCG@20 or MRR, our results suggest that the sample should contain no less than 1,500 documents, and be created using a document representation that considers anchor text in addition to the content of the document, so as to ensure effective retrieval for both informational and navigational information needs (Sect. 5.1.4). If a listwise learning to rank technique is used to obtain the learned model, then our results suggest that NDCG@10 represents a suitably informative learning evaluation measure to achieve an effective learned model (Sect. 5.2.3). Lastly, the importance of different classes of features within a learned model are dependent on both the sample size and the rank cutoff of the learning evaluation measure (Sect. 5.3.3).

This work used a total of six learning to rank techniques that are widely implemented or freely available, which are representative of the various families (pointwise, pairwise and listwise), as well as of different types of learned models (linear combination, neural network, or regression tree). The used learning to rank techniques includes the state-of-the-art LambdaMART technique, which won the Yahoo! 2011 learning to rank challenge (Chapelle and Chang 2011). While some other learning to rank techniques have been proposed in the literature [as reviewed by Liu (2009)], the breadth of those learning techniques that we experimented with and their wide deployment in the literature and public evaluation forums leads us to strongly believe that our results should generalise to the learned models obtained from other learning to rank techniques.

**Acknowledgments** We thank the reviewers and the Co-Editor-in-Chief, Josiane Mothe, for their useful feedback. Part of this work has been carried out in the scope of the EC co-funded project SMART (FP7-287583). We also acknowledge and thank the authors of the open source learning to rank tools that were used in the experiments of this article.

## References

Amati, G. (2003).*Probabilistic models for information retrieval based on divergence from randomness*. PhD thesis, Department of Computing Science, University of Glasgow.

Amati, G., Ambrosi, E., Bianchi, M., Gaibisso, C., & Gambosi, G. (2008). FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog Track. In *Proceedings of the 16th text retrieval conference, TREC '07*.

Arampatzis, A., Kamps, J., & Robertson, S. (2009). Where to stop reading a ranked list?: Threshold optimization using truncated score distributions. In *Proceedings of the 32nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '09* (pp. 524–531). doi:10.1145/1571941.1572031.

Aslam, J. A., Kanoulas, E., Pavlu, V., Savev, S., & Yilmaz, E. (2009). Document selection methodologies for efficient and effective learning-to-rank. In *Proceedings of the 32nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '09* (pp. 468–475). doi:10.1145/1571941.1572022.

Becchetti, L., Castillo, C., Donato, D., Leonardi, S., & Baeza-Yates, R. (2006). Link-based characterization and detection of Web spam. In: *Proceedings of the 2nd international workshop on adversarial information retrieval on the web, AIRWeb*.

Beitzel, S. M., Jensen, E. C., Chowdhury, A., Grossman, D., Frieder, O., & Goharian, N. (2004). Fusion of effective retrieval strategies in the same information retrieval system. *Journal American Society of Information Science & Technology, 55*(10), 859–868. doi:10.1002/asi.20012.

Broder, A. Z., Carmel, D., Herscovici, M., Soffer, A., & Zien. J. (2003). Efficient query evaluation using a two-level retrieval process. In *Proceedings of the 12th ACM international conference on information and knowledge management, CIKM '03* (pp. 426–434). doi:10.1145/956863.956944.

Buckley, C., & Voorhees, E. M. (2000). Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '00* (pp. 33–40). doi:10.1145/345508.345543.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on machine learning, ICML '05* (pp. 89–96). doi:10.1145/1102351.1102363.

Cambazoglu, B. B., Zaragoza, H., Chapelle, O., Chen, J., Liao, C., Zheng, Z., & Degenhardt, J. (2010). Early exit optimizations for additive machine learned ranking systems. In *Proceedings of the third ACM international conference on web search and data mining, WSDM '10* (pp. 411–420). doi:10.1145/1718487.1718538.

Carterette, B., Fang, H., Pavlu, V., & Kanoulas, E. (2010). Million query track 2009 overview. In *Proceedings of the 18th text retrieval conference, TREC '09*.

Castillo, C., Donato, D., Becchetti, L., Boldi, P., Leonardi, S., Santini, M., & Vigna, S. (2006). A reference collection for web spam. *SIGIR Forum, 40*(2), 11–24. doi:10.1145/1189702.1189703.

Chapelle, O., & Chang, Y. (2011). Yahoo! learning to rank challenge overview. In: *Journal of Machine Learning Research Proceedings Track, 14*, 1–24.

Chapelle, O., Metlzer, D., Zhang, Y., & Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceeding of the 18th ACM international conference on information and knowledge management, CIKM '09* (pp. 621–630). doi:10.1145/1645953.1646033.

Chapelle, O., Chang, Y., & Liu, T. Y. (2011). Future directions in learning to rank. *Journal of Machine Learning Research Proceedings Track, 14*, 91–100.

Clarke, C. L. A., Craswell, N., & Soboroff, I. (2010). Overview of the TREC 2009 Web track. In *Proceedings of the 18th text retrieval conference (TREC 2009), TREC '09*.

Clarke, C. L. A., Craswell, N., & Soboroff, I. (2011). Overview of the TREC 2010 web track. In *Proceedings of the 19th text retrieval conference, TREC '10*.

Coolican, H. (1999). *Research methods and statistics in psychology*. London: A Hodder Arnold Publication, Hodder & Stoughton. http://books.google.co.uk/books?id=XmfGQgAACAAJ.

Cormack, G. V., Smucker, M. D., & Clarke, C. L. A. (2011). Efficient and effective spam filtering and re-ranking for large Web datasets. *Information Retrieval, 15*(5), 441–465. doi:10.1007/s10791-011-9162-z.

Craswell, N., & Hawking, D. (2004). Overview of TREC-2004 web track. In *Proceedings of the 13th text retrieval conference, TREC '04*.

Craswell, N., Robertson, S., Zaragoza, H., & Taylor, M. (2005). Relevance weighting for query independent evidence. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '05* (pp. 416–423). doi:10.1145/1076034.1076106.

Craswell, N., Jones, R., Dupret, G., & Viegas, E. (eds) (2009). *Proceedings of the 2009 workshop on web search click data*. doi:10.1145/1507509.

Craswell, N., Fetterly, D., Najork, M., Robertson, S., & Yilmaz, E. (2010). Microsoft research at TREC 2009. In *Proceedings of the 18th Text REtrieval Conference, TREC '09*.

Croft, W. B. (2008). Learning about ranking and retrieval models. In *Keynote, SIGIR 2007 workshop learning to rank for information retrieval (LR4IR)*.

Donmez, P., & Carbonell, J. G. (2009). Active sampling for rank learning via optimizing the area under the roc curve. In *Proceedings of the 31th European conference on IR research on advances in information retrieval, ECIR '09* (pp. 78–89). doi:10.1007/978-3-642-00958-7_10.

Donmez, P., Svore, K. M., & Burges, C. J. (2009). On the local optimality of LambdaRank. In: *Proceedings of the 32nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '09* (pp. 460–467). doi:10.1145/1571941.1572021.

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research, 4*, 933–969.

Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*, 1189–1232. doi:10.1214/aos/1013203451.

Ganjisaffar, Y., Caruana, R., & Lopes, C. (2011). Bagging gradient-boosted trees for high precision, low variance ranking models. In: *Proceedings of the 34th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '11* (pp. 85–94). doi:10.1145/2009916.2009932.

Hawking, D., Upstill, T., & Craswell, N. (2004). Toward better weighting of anchors. In: *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '04* (pp. 512–513). doi:10.1145/1008992.1009096.

He, B., Macdonald, C., & Ounis, I. (2008). Retrieval sensitivity under training using different measures. In: *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '08* (pp. 67–74). doi:10.1145/1390334.1390348.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *Transactions on Information Systems, 20*(4), 422–446. doi:10.1145/582415.582418.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. doi:10.1126/science.220.4598.671.

Kraaij, W., Westerveld, T., & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '02* (pp. 27–34). doi:10.1145/564376.564383.

Li, H. (2011). *Learning to rank for information retrieval and natural language processing. Synthesis lectures on human language technologies*. San Rafael: Morgan & Claypool Publishers. doi:10.2200/S00348ED1V01Y201104HLT012.

Liu, T. Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval, 3*(3), 225–331. doi:10.1561/1500000016.

Long, B., Chapelle, O., Zhang, Y., Chang, Y., Zheng, Z., & Tseng. B. (2010). Active learning for ranking through expected loss optimization. In: *Proceedings of the 33rd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '10* (pp. 267–274). doi: 10.1145/1835449.1835495.

Macdonald, C., & Ounis, I. (2009). Usefulness of quality click-through data for training. In: *Proceedings of the 2009 workshop on web search click data, WSCD '09* (pp. 75–79). doi:10.1145/1507509.1507521.

Metzler, D. (2007). Automatic feature selection in the Markov random field model for information retrieval. In: *Proceedings of the 16th ACM international conference on information and knowledge management, CIKM '07* (pp. 253–262). doi:10.1145/1321440.1321478.

Metzler, D., & Croft, W. B. (2005). A Markov random field model for term dependencies. In: *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '05* (pp. 472–479). doi:10.1145/1076034.1076115.

Minka, T., & Robertson, S. (2008). Selection bias in the LETOR datasets. In: *SIGIR 2007 workshop learning to rank for information retrieval (LR4IR).*

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., & Lioma, C. (2006). Terrier: A high performance and scalable information retrieval platform. In: *Proceedings of the 2nd workshop on open source information retrieval at SIGIR 2006, OSIR* (pp. 18–25).

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The pagerank citation ranking: Bringing order to the web*. Technical report, Stanford Digital Library technologies project.

Pederson, J. (2008). The machine learned ranking story. http://jopedersen.com/Presentations/The_MLR_Story.pdf, Accessed July 30, 2012.

Pederson, J. (2010). Query understanding at bing. In: *Invited talk, SIGIR 2010 industry day*.

Peng, J., Macdonald, C., He, B., Plachouras, V., & Ounis, I. (2007). Incorporating term dependency in the DFR framework. In: *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '07* (pp. 843–844). doi:10.1145/1277741.1277937.

Piroi, F., & Zenz, V. (2011). Evaluating information retrieval in the intellectual property domain: The CLEF-IP campaign. In *Current challenges in patent information retrieval, the information retrieval series, 29* (pp. 87–108). Berlin: Springer. doi:10.1007/978-3-642-19231-9_4.

Plachouras, V. (2006) *Selective web information retrieval. PhD thesis*. Department of Computing Science, University of Glasgow.

Plachouras, V., & Ounis, I. (2004). Usefulness of hyperlink structure for query-biased topic distillation. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '04* (pp. 448–455). doi:10.1145/1008992.1009069.

Plachouras, V., Ounis, I., & Amati, G. (2005). The static absorbing model for the web. *Journal of Web Engineering, 4*(2), 165–186.

Qin, T., Liu, T. Y., Xu, J., & Li, H. (2009). LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval, 13*(4), 347–374.

Robertson, S. (2008). On the optimisation of evaluation metrics. In *Keynote, SIGIR 2008 workshop learning to rank for information retrieval (LR4IR).*

Robertson, S., & Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M. (1992). Okapi at TREC. In *Proceedings of the text retrieval conference, TREC-1.*

Segalovich, I. (2010). Machine learning in search quality at Yandex. In *Invited Talk, SIGIR 2010 industry day.*

Tomlinson, S., & Hedin, B. (2011). Measuring effectiveness in the TREC legal track. In *Current challenges in patent information retrieval, the information retrieval series* (vol. 29, pp. 167–180). Berlin: Springer. doi:10.1007/978-3-642-19231-9_8.

Voorhees, E. M., & Harman, D. K. (2005). *TREC: Experiment and evaluation in information retrieval.* Cambridge: MIT Press. doi:10.1002/asi.20583.

Weinberger, K., Mohan, A., & Chen, Z. (2010). Tree ensembles and transfer learning. In *Proceedings of the Yahoo! learning to rank challenge workshop at WWW 2010.*

Wu, Q., Burges, C. J. C., Svore, K. M., & Gao, J. (2008). *Ranking, boosting, and model adaptation.* Technical Report MSR-TR-2008-109, Microsoft.

Xu, J., & Li, H. (2007). Adarank: A boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '07* (pp. 391–398). doi:10.1145/1277741.1277809.

Yilmaz, E., & Robertson, S. (2010). On the choice of effectiveness measures for learning to rank. *Information Retrieval, 13*(3), 271–290. doi:10.1007/s10791-009-9116-x.

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '01* (pp. 334–342). doi:10.1145/383952.384019.

Zhang, M., Kuang, D., Hua, G., Liu, Y., & Ma, S. (2009). Is learning to rank effective for web search? In *Proceedings of SIGIR 2008 workshop learning to rank for information retrieval (LR4IR)*.

Zobel, J. (1998). How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '98* (pp. 307–314). doi:10.1145/290941.291014.