Content-Based Relevance Estimation in Retrieval Settings with Ranking-Incentivized Document Manipulations

Ziv Vasilisky
ziv.vasilisky@mktmediastats.com
MKT MEDIASTATS
Israel

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

Moshe Tennenholtz
moshet@ie.technion.ac.il
Technion
Israel

Fiana Raiber
fiana@yahooinc.com
Yahoo Research
Israel

ABSTRACT

In retrieval settings such as the Web, many document authors are ranking incentivized: they opt to have their documents highly ranked for queries of interest. Consequently, they often respond to rankings by modifying their documents. These modifications can hurt retrieval effectiveness even if the resultant documents are of high quality. We present novel content-based relevance estimates which are "ranking-incentives aware"; that is, the underlying assumption is that content can be the result of ranking incentives rather than of pure authorship considerations. The suggested estimates are based on inducing information from past dynamics of the document corpus. Empirical evaluation attests to the clear merits of our most effective methods. For example, they substantially outperform state-of-the-art approaches that were not designed to address ranking-incentivized document manipulations.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Retrieval models and ranking**;

KEYWORDS

competitive retrieval; language modeling; learning-to-rank

ACM Reference Format:
Ziv Vasilisky, Oren Kurland, Moshe Tennenholtz, and Fiana Raiber. 2023. Content-Based Relevance Estimation in Retrieval Settings with Ranking-Incentivized Document Manipulations. In *Proceedings of the 2023 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '23), July 23, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3578337.3605124

1 INTRODUCTION

Many authors of Web documents are incentivized to have their documents highly ranked for queries of interest [3]. Top-ranked documents attract most user attention and engagement [19] which is important, for example, for queries with commercial intent.

As a result of their ranking incentives, authors may respond to rankings induced by the search engine by modifying their documents; the goal is to promote the documents in future rankings. Thus, there are queries for which there is essentially a ranking competition [24, 49]. Accordingly, such search settings were recently termed: "*competitive search settings*" [24], with the Web being a canonical example. Herein, we refer to a search setting with ranking incentivized document manipulations as a "competitive search setting". A search setting where authors have no clear ranking incentives (e.g., a library of books) is termed a "*non-competitive search setting*".

Ranking incentivized modifications are often referred to as search engine optimization (SEO) [13, 28]. Our focus in this paper, as in some recent line of work [9–11, 40, 46, 48, 51, 53], is on content modifications which are the result of ranking incentives. A well known content modification approach intended to promote documents in rankings is keyword stuffing [13]: adding the terms of a query for which rank promotion is desired to the document. Other examples of manipulation strategies are the increased or reduced use of stopwords [40] and the substitution of terms with their synonyms [46, 48, 51, 53]. Manipulation can be well beyond addition or substitution of several terms. As a case in point, the topical focus of a document can change to potentially attain improved ranking [11, 40]. The document length can also significantly change [11].

Ranking-incentivized content manipulation strategies as those just mentioned can hurt retrieval effectiveness for both classical and neural retrieval methods [40, 54]. For example, in Okapi BM25, a document retrieval score is easily increased by stuffing query terms. If the document is not relevant, the score increase does not necessarily reflect increased level of relevance.

Ranking incentivized document manipulations do not necessarily hurt the document quality (e.g., its textual coherence), and more specifically, do not necessarily turn it into spam [10, 38, 40]. Hence, using spam classifiers and more generally document quality estimates, which is standard practice in work on Web retrieval [2], can fall short in addressing ranking-incentivized content manipulations [40]. Indeed, as Jones et al. [20] noted: "*not all content that complicates ranking is also spam*".

Content manipulation can potentially be addressed by devising retrieval mechanisms that tackle specific manipulation strategies;

ICTIR '23, July 23, 2023, Taipei, Taiwan

ACM ISBN 979-8-4007-0073-6/23/07…$15.00
https://doi.org/10.1145/3578337.3605124

e.g., substitution of terms with their synonyms [52]. However, as noted above, there is a wide array of manipulation strategies. Furthermore, it was observed that various strategies are often employed simultaneously [11, 40].

All these observations led us to pursue the following challenge in this paper: *devising content-based relevance estimates that are "aware" of ranking incentives and which are not "tailored" to specific manipulation strategies*. Awareness to ranking incentives means making the assumption that content in documents might be the result of ranking-incentivized manipulations rather than of pure authorship considerations.

To address the research challenge, we propose a suite of methods that utilize information induced from past dynamics — namely, content changes — of the corpus. For example, significant change of occurrence of specific terms in a document along time, or continuous topical shift, can potentially attest to ranking-incentivized manipulations. We pay particular attention to the following research questions. The first touches on the fundamental difference between competitive and non-competitive retrieval settings:

RQ1: Can ranking-incentives aware content-based relevance estimates outperform existing relevance estimates which were not devised specifically for competitive retrieval settings with ranking-incentivized content manipulations?

The second and third research questions touch on the differences, and potentially complementary nature, of addressing content manipulations that tend to degrade document quality and manipulations which do not:

RQ2: Is using ranking-incentives aware relevance estimates of merit with respect to using (query-independent) document quality estimates (e.g., spam classification)?

RQ3: Is integrating ranking-incentives aware relevance estimates and document quality estimates of merit?

We explore two major retrieval frameworks for devising content-based retrieval estimates that are ranking-incentives aware and which are based on utilizing information induced from past corpus dynamics. Our first approach operates in the language-modeling framework to retrieval [5]. We present a novel generative assumption to content creation in documents in competitive retrieval settings. Namely, one of the content generators in a document is presumed to be driven by ranking incentives. Using the assumption we derive a novel document language model. We note that recent work showed that a standard simple language-model-based approach [35, 56] posts retrieval performance that is better, and yields rankings that are more robust, than that of neural retrieval methods in a competitive retrieval setting with ranking-incentivized manipulations [54]. Our second approach operates in the feature-based learning-to-rank framework [27] with suggested features that quantify past temporal changes of existing content-based features.

For empirical evaluation, we used recently published datasets [10, 40]. These are recordings of iterative content-based relevance ranking competitions held between ranking-incentivized authors. The evaluation demonstrates the merits of our most effective methods. Using our novel language model results in performance that consistently surpasses that of the standard language model approach, which as noted above, is highly effective for competitive retrieval settings [54]. Furthermore, using features that quantify past changes of document content results in retrieval performance that transcends the state-of-the-art in content-based relevance estimation using feature-based learning-to-rank [27] (RQ1). In addition, our learning-to-rank approach outperforms *manual* filtering of documents of low quality (RQ2), and is also effective in addition to using such filtering (RQ3). These findings provide support to (i) the potential merit of accounting for ranking incentives in devising relevance estimates for competitive retrieval settings, and to the (ii) difference between, and complementary nature of, our ranking-incentives aware relevance estimates and query-independent document quality estimates that are mainly used to penalize low quality documents.

Our contributions can be summarized as follows:

- The first work, to the best of our knowledge, on content-based relevance estimates that address ranking-incentivized content manipulations in competitive retrieval settings from a "macro-level" perspective; that is, without being tailored to a specific type of content manipulation strategy.
- Our novel relevance estimates are shown to outperform state-of-the-art estimates which were not designed to address ranking-incentivized content manipulation.
- We demonstrate the performance superiority of using our ranking-incentives aware relevance estimates to document quality measures and the merits in integrating them.

2 RELATED WORK

The focus of earlier work on adversarial information retrieval [3] has been on detecting different types of spam and making hyperlink-based models more robust to adversarial effects [3, 14, 32].

Various aspects of temporal retrieval were studied [21]; e.g., temporal indexing techniques, retrieval models for temporal queries, or having time as an additional "relevance dimension". In contrast, we leverage past temporal corpus dynamics for content-based relevance estimation in competitive retrieval settings.

There is a huge body of work on analyzing and predicting temporal changes of Web documents regardless of ranking effects; e.g., [36, 37, 44]. Our approaches utilize information induced from temporal changes to devise relevance estimates.

Raifer et al. [40] studied the document modification strategies of authors in ranking games. We use the dataset of the controlled ranking competitions they organized. However, no retrieval method was proposed in their work.

A game theoretic analysis was used to show that the probability ranking principle (PRP) [42] is sub-optimal in competitive retrieval settings. A theoretically improved stochastic ranking paradigm was proposed but it relies on true relevance judgments. In contrast, we devise relevance estimates that do not use relevance judgments.

There is a line of work on using past snapshots of a document — mainly, term and document frequencies — to enrich its representation [1, 7, 8, 33]. Competitive retrieval settings were not considered

and therefore the effect of ranking incentives on document manipulations was not accounted for. Still, we use a couple of these approaches [1, 8] as baselines as they utilize information induced from past snapshots of the corpus. We show that our approaches substantially outperform these baselines.

Raiber et al. [38] addressed the task of predicting whether a query would be the target for SEO. In contrast to our work, the document ranking task was not addressed. Raiber et al. [38] also showed using human annotations that documents which went through SEO could be ranked high for TREC's ClueWeb09 queries even if various document quality measures [2] were used by the ranking function. We demonstrate the merits of our ranking-incentives aware relevance estimates even when applied after documents marked by humans as of low quality are filtered out.

There is work on filtering from ranked lists documents that exhibit high surface-level query similarity but for which other relevance evidence — induced from inter-document similarities — is not significant [39]. The premise is that these documents might have been manipulated by addition of query terms. The filtering approach was substantially inferior to a standard relevance estimation method that simply used these inter-document similarities [39]. In contrast to our work, past corpus snapshots were not utilized and a ranking-incentives aware retrieval model was not proposed.

Recent work demonstrated the connection between the robustness to document modifications of rankings induced by feature-based learning-to-rank methods in competitive retrieval settings and the level of regularization applied to these methods [9]. The relevance effects of modifications were not studied, and improved relevance estimates for competitive retrieval settings were not proposed. Ranking robustness, which is outside the scope of this paper, was also empirically studied for neural retrieval methods [54].

There is work on automatically augmenting documents with phrases [17], replacing their passages [10], or substituting individual terms [41, 46, 48, 51, 53] to promote documents in rankings. There is also work on defense mechanisms against term-substitution attacks on rankers [52]. Our relevance estimates are not "tailored" to specific manipulation strategies. As noted above, there are various manipulation strategies which are often employed simultaneously as is the case in the datasets we use for evaluation.

3 RETRIEVAL FRAMEWORKS

We address the task of ad hoc retrieval over a document corpus $\mathcal{D}$. We assume that some of the temporal *content dynamics* in the corpus is driven by ranking incentivized document authors: they modify their documents in response to rankings induced for queries of interest to potentially improve the documents' future rankings.

Our approach is based on utilizing information induced from the dynamics of corpus changes to improve the effectiveness of retrieval over $\mathcal{D}$. Let $\mathcal{D}_{-i}$ ($i \in \{1, \ldots, h\}$) denote the i'th historical (previous) snapshot of $\mathcal{D}$; h is the history length; we will sometimes use $\mathcal{D}_0$ to refer to $\mathcal{D}$. If d is a document in $\mathcal{D}$, we use d_{-i} to denote its i'th past version which is part of $\mathcal{D}_{-i}$; $d \equiv d_0$.

Throughout this section we assume a fixed query q and some document relevance ranking method $\mathcal{M}$. We assume that $\mathcal{M}$ was used to induce rankings for q over each of $\mathcal{D}_0, \ldots, \mathcal{D}_{-h}$. A competitive retrieval setting means that document authors might have responded to the ranking induced over $\mathcal{D}_{-i}$ ($i \in 1, \ldots, h$), specifically by modifying their documents with the goal of improving their future ranking — i.e., that induced over $\mathcal{D}_{-i+1}$.

We set as a goal to devise methods that utilize information induced from past temporal content dynamics of the corpus so as to address potential effects of ranking-incentivized document modifications. We pursue our goal in two major *retrieval frameworks*. The first, addressed in Section 3.1, is the language modeling framework to retrieval [5]. The second framework we explore (Section 3.2) is feature-based learning-to-rank [27].

3.1 The Language Modeling Framework

We now present a retrieval approach in the language modeling framework that addresses content effects driven by ranking incentives.

Most document ranking methods in the language modeling framework to retrieval [5, 25] are based on comparing a language model induced from a document with that induced from the query[1]. Unigram document language models [5] — based on a term independence assumption — are often estimated using term counts in the document and in the corpus as we formally describe below. A generative perspective of mixing these two types of counts is that document terms are generated by a two component mixture model [15, 29]. The first component is the '"core/pure" authorship model (e.g., a topical model) and the second is a general (background) model of language approximated by that induced from the corpus.

In competitive retrieval settings, document content can result from ranking incentives. Hence, we make the generative assumption that in competitive settings, terms in documents are generated by a three component mixture model: (i) the "core" model, (ii) a ranking-incentives aware model, and (iii) the background model. The ranking-incentives aware model, which could be viewed as an adversarial model with respect to the ranking function, is query dependent. That is, content generation is affected by queries for which the document author has ranking incentives. Hence, our generative assumption to document creation departs from previous work on devising document language models for retrieval in two important respects: it accounts for ranking-incentives effects and it assumes that document creation might be query dependent. We now turn to formally present the mixture model where the goal is to estimate the "core" document model. That is, we aim to rank the document based on its "real" content and neutralize the effects of content created for the sake of rank promotion.

3.1.1 Mixture Model. We first describe the notation that will be used throughout this section. Let

$$p_x^{MLE}(w) \stackrel{def}{=} \frac{\text{tf}(w \in x)}{\sum_{w'} \text{tf}(w' \in x)} \quad (1)$$

be the maximum likelihood estimate (MLE) of term w with respect to text (or text collection) x; $\text{tf}(w \in x)$ is the number of occurrences of w in x. We smooth the MLE using Dirichlet priors [56]:

$$p_x^{Dir}(w) \stackrel{def}{=} (1 - \alpha_x) p_x^{MLE}(w) + \alpha_x p_{\mathcal{D}}^{MLE}(w); \quad (2)$$

[1] Using an unsmoothed maximum likelihood estimate for the query model, and the KL divergence to compare document and query language models, results in rank equivalence to the query likelihood model [47].

$\alpha_x \stackrel{def}{=} \frac{\mu}{|x|+\mu}$ where $|x| \stackrel{def}{=} \sum_{w' \in x} \text{tf}(w' \in x)$ and μ is a free parameter.

We use the cross entropy to compare two language models θ_1 and θ_2: $CE(\theta_1 \mid\mid \theta_2) \stackrel{def}{=} -\sum_w p(w|\theta_1) \log p(w|\theta_2)$.

Let d be a document in $\mathcal{D}$. Recall that q is the given query for which ranking should be induced. As noted above, for the competitive retrieval setting we assume that the terms in d are generated by a three component mixture model of unigram language models. The first component is a latent "core" model, $p_d^{core}(\cdot)$, which should be estimated. The second component is the ranking-incentives aware model, $p_q^{rinc}()$. This query-dependent model assigns high probability to terms that are presumably likely to be used by authors competing for high ranking with respect to q. We describe below a few approaches to estimating $p_q^{rinc}(\cdot)$. The third component in the mixture is the background language model induced from the corpus: $p_{\mathcal{D}}^{MLE}(\cdot)$.

Accordingly, d's log generation likelihood can be written as:

$$\mathcal{L}(d) = \log p(d|q, p_d^{core}(\cdot)) \stackrel{def}{=} \sum_{w_i \in d} \log[(1-\lambda_1-\lambda_2)p_d^{core}(w_i) + \lambda_1 p_q^{rinc}(w_i) + \lambda_2 p_{\mathcal{D}}^{MLE}(w_i)]; \quad (3)$$

w_i is the *i'th term* in d; λ_1 and λ_2 are free parameters. We estimate $p_d^{core}(\cdot)$ using the EM algorithm. The EM update steps are (w is *a term* in d):

$$\text{E: } f(w) = \frac{(1-\lambda_1-\lambda_2)p_d^{core}(w)}{(1-\lambda_1-\lambda_2)p_d^{core}(w) + \lambda_1 p_q^{rinc}(w) + \lambda_2 p_{\mathcal{D}}^{MLE}(w)};$$

$$\text{M: } p_d^{core}(w) = \frac{\text{tf}(w \in d) f(w)}{\sum_{w'} \text{tf}(w' \in d) f(w')}.$$

We cannot use $p_d^{core}(\cdot)$ to directly rank d as it might assign a zero probability to terms not in d. Hence, we smooth $p_d^{core}(\cdot)$ similarly to the smoothing of maximum likelihood estimates in Equation 2:

$$p_d^{core;Dir}(w) \stackrel{def}{=} (1-\alpha_d)p_d^{core}(w) + \alpha_d p_{\mathcal{D}}^{MLE}(w); \quad (4)$$

setting $\alpha_d \stackrel{def}{=} \frac{\mu}{|d|+\mu}$, with a free parameter μ, yields a variant of Dirichlet smoothing[2].

We then assign document d a retrieval score for query q:

$$s(q;d) \stackrel{def}{=} -CE(p_q^{MLE}(\cdot) \mid\mid p_d^{core;Dir}(\cdot)). \quad (5)$$

A ranking-incentives aware model. To estimate a ranking-incentives aware model for query q, $p_q^{rinc}(\cdot)$, we use a document set $\mathcal{D}_q^{rinc}$ composed of documents that are suspected to manifest ranking-incentives effects with respect to q; e.g., excessive use of q's terms. We then set for each term w: $p_q^{rinc}(w) \stackrel{def}{=} p_{\mathcal{D}_q^{rinc}}^{MLE}(w)$. See Equation 1 for a definition of the MLE.

We consider two sets of documents, $\mathcal{D}_q^{rinc}$, which might manifest ranking-incentives effects for q. The first is the set of documents most highly ranked in the past for q. Let $d_{q;\mathcal{D}_{-i}}^{top}$ denote the document most highly ranked for q in $\mathcal{D}_{-i}$: the i'th historical snapshot of the corpus $\mathcal{D}$. We define the **TopRank** set: $\mathcal{D}_{q,TopRank}^{rinc} \stackrel{def}{=} \{d_{q;\mathcal{D}_{-1}}^{top}, \ldots, d_{q;\mathcal{D}_{-k}}^{top}\}$; k is a free parameter. The assumption is that in a competitive retrieval setting, these documents should manifest to some extent ranking-incentives effects. For example, Raifer et al. [40] showed empirically that document authors try to promote their documents by mimicking content of documents that were highly ranked for the query in the past. Theoretical analysis provided the motivation for this strategy [40]: highly ranked documents are a signal about the undisclosed ranking function.

The second set of documents that might manifest ranking incentives effects, referred to as **HighImp**, is composed of past versions $d_{-1}, \ldots, d_{-k+1}$ (k is a free parameter) of a document d for which the rank improvement of moving from d_{-k} (in the corpus snapshot $\mathcal{D}_{-k}$) to d_{-1} (in the corpus snapshot $\mathcal{D}_{-1}$) was the highest with respect to all rank improvements for documents between these two corpus snapshots; if two documents had the same rank improvement between $\mathcal{D}_{-k}$ and $\mathcal{D}_{-1}$, we use the one whose rank in the ranking induced over $\mathcal{D}_{-1}$ was higher. Significant rank promotion between rankings induced for a query over the historical corpus snapshots can potentially be the result of increased attempt at manipulating the document to this end.

3.2 The Learning-To-Rank Framework

Our proposed mixture model is an example in the language modeling framework of accounting for the potential effects of a competitive retrieval setting where document modifications can be ranking incentivized. Information induced from historical corpus snapshots served as the basis to model these potential effects. We now turn to explore the use of such information to improve an existing feature-based learning to rank (LTR) approach which operates in a competitive retrieval setting.

Suppose that the retrieval method $\mathcal{M}$ used to induce the rankings for q over the corpus snapshots $\mathcal{D}_{-h}, \ldots, \mathcal{D}_{-1}, \mathcal{D}_0 \equiv \mathcal{D}$ is feature-based [27]; that is, each pair of a query and a document is represented using a feature vector and $\mathcal{M}$ is an LTR approach. Let $\mathcal{F}$ denote the subset of features used by $\mathcal{M}$ which are based on textual content in the document; e.g., Okapi BM25 document-query similarity, document length, number of query terms which appear in the document, the entropy of the term distribution in the document [2], etc. We use f to denote a feature in $\mathcal{F}$ and $v_f(q,d)$ to denote its value for a query-document pair (q,d).

We consider a set of additional features, referred to as **Agg**, to be added to the base feature set $\mathcal{F}$. The Agg set is composed of features that are aggregates of historical values of features in $\mathcal{F}$. Specifically, for each feature f ($\in \mathcal{F}$) and a query-document pair (q,d), we use in addition to $v_f(q,d)$ features whose values are the mean (**Avg**), maximum (**Max**), minimum (**Min**) and standard deviation (**Std**) of the feature values: $\{v_f(q,d_{-i})\}_{i=1}^{h}$. The resultant features are denoted f-x where x $\in$ {Avg, Max, Min, Std}.

These features help to quantify, for example, temporal skewness. For query-document similarity features, high skewness towards the present might attest to increased effort of manipulating the document for improved future ranking. Hence, the learned ranking function might assign reduced importance to features with high historical skweness of values. While our approach focuses on modeling temporal changes of content-based features, the aggregation-based features we present are not committed to content.

[2] We write "variant" as $p_d^{core}(\cdot)$ is not an MLE for a multinomial distribution for which Dirichlet is a conjugate prior.

Table 1: Datasets used for evaluation.

	ASRC	Combined
overall # documents	1279	1090
# queries	31	31
# documents per query and round	5-6	7-8
the rounds used for evaluation	2-8	2-5
% of relevant documents	87%	88%
% of documents with relevance grade 1	12%	11.1%
% of documents with relevance grade 2	23.4%	22.6%
% of documents with relevance grade 3	51.7%	54.5%
% of low quality documents	0.9%	1.4%

We found that using estimates of the changes of the similarity of a document d to its past snapshots, $\{d_{-i}\}_{i=1}^{h}$, can help to further improve retrieval effectiveness. For example, high variance of the similarity can potentially attest to excessive changes of the document which might be driven by ranking incentives. The inter-document similarity is measured using the cosine between tf.idf vectors. We use the average, maximum, minimum and standard deviation of the similarity of d with its past versions, $\{d_{-i}\}_{i=1}^{h}$, as the features: Sim-Avg, Sim-Max, Sim-Min and Sim-Std. These features are added to the Agg feature set.

We use **LTR** to denote the $\mathcal{M}$ retrieval method applied with its original set of features. **LTR+Agg** refers to using the features in Agg in addition to the original features.

4 EXPERIMENTAL SETTING

We next describe the datasets used for evaluation. Then, we provide details about the learning-to-rank approach, the reference comparisons used, and additional aspects of the evaluation: evaluation metrics, cross validation and ranges of free-parameter values.

4.1 Data

Our retrieval methods address the content-based corpus dynamics driven by ranking incentives of authors. Hence, the evaluation requires data that manifests such ranking-incentivized dynamics. Furthermore, the data should include historical snapshots of a corpus and rankings induced over these snapshots for various queries.

To the best of our knowledge, there are only two publicly available datasets that meet the requirements just specified. These datasets are the recordings of content-based ranking competitions organized by Raifer et al. [40][3] and Goren et al. [10][4]. The competitions were held between students in courses. The students manipulated plain-text documents of up to 150 terms. The competitions were iterative and ran for a few rounds. At the beginning of the competitions, for each query the students were provided with the *same* example of a relevant document [10, 40]. In each round the students were shown a ranking induced over their documents by an undisclosed ranking function: LambdaMART [55] with content-based features [10, 40]. Then, the students modified their documents to potentially improve their rankings in the next round. The students were incentivized via bonuses to course grades to participate in the competitions. Ethics committees approved these competitions [10].

Raifer et al.'s [40] competitions were held for 31 queries (topic titles) from TREC's ClueWeb09 collection [38]. Each competition for a query was run for 8 rounds between 5 students except for one which had 6 students. We use *each* of the 2–8 rounds as an evaluation setting: the corpus $\mathcal{D}$ ($\equiv \mathcal{D}_0$) in round i ($\in \{2, \ldots, 8\}$) is used for evaluation; its historical snapshots are: $\mathcal{D}_{-1}, \ldots, \mathcal{D}_{-h}$ where $\mathcal{D}_{-j}$ ($j \in \{1, \ldots h\}$) is the corpus snapshot in round $i - j$; $h = i - 1$. This dataset was termed **ASRC**.

The competitions of Goren et al. [10] were run for 5 rounds with 30 queries out of the 31 used by Raifer et al. [40][5]. Only two students competed in each competition for a query; the other competing documents were planted. Since ranking two documents results in a somewhat unstable evaluation, we combined for each query and each round in $\{1, \ldots, 5\}$ the documents of Goren et al. [10] with those of Raifer et al. [40] from the respective round[6]. The corpora used for evaluation are for each of the rounds 2–5. The resultant dataset is denoted **Combined**.

Each document in the datasets of Raifer et al. [40] and Goren et al. [10] was judged for binary relevance to the queries by five annotators. We induced graded relevance judgments as follows: 0 grade (non-relevant document), if less than 3 annotators deemed the document relevant; otherwise, $x - 2$ grade (relevant document), where x (≥ 3) is the number of annotators who deemed the document relevant. Each document in the ASRC dataset was labeled by five annotators as whether it was keyword-stuffed [40]. That is, whether the document contained an excessive use of specific terms. Keyword stuffing is a common search engine optimization technique [13]. If three or more annotators marked it as keyword stuffed, and its relevance grade was lower than 2, then we consider it as of *low quality*. Similarly, three annotators labeled each document in Goren et al.'s dataset [10] as of high quality or low quality. If a document was marked by all three as of low quality, and its relevance grade was lower than 2, we treat it as of low quality[7].

The details of the two datasets are summarized in Table 1. Obviously, these are not big datasets, but they have already been used by several research groups for (i) analyzing document modification strategies in competitive retrieval settings [40], (ii) exploring ranking robustness of both classical and neural retrieval methods in these settings [9, 54], and (iii) evaluation of automatic methods of content modification for rank promotion [10]. Furthermore, it was recently reported [50] that document modification patterns observed in these ranking competitions [40] were also observed with respect to past versions of ClueWeb09 documents found in the Internet Archive. We could not use this dataset [50] due to the extreme sparseness of past snapshots of the corpus; many documents in ClueWeb09 did not have even one past snapshot. Organizing large-scale ranking competitions that will result in much larger datasets is a significant challenge left for future work.

Table 1 shows that the percentage of relevant documents is high. This is because the students opted (although were not explicitly instructed) to produce relevant documents. Still, there is a solid

[3] https://asrcdataset.github.io/asrc/.

[4] https://github.com/asrcompetition/content_modification_dataset/tree/master/ControledExperiment.

[5] We used the control competitions data only. Other competitions involved automatic manipulation of documents by bots.

[6] For the query used in Raifer et al. [40] but not in Goren et al. [10] we used only the rounds and documents of the former.

[7] Some of the highly relevant documents in Raifer et al. [40] were marked as keyword stuffed and some of the highly relevant documents in Goren et al. [10] were marked as of low quality. Here, we address low quality documents with a low relevance grade.

percentage of documents with a relatively low relevance grade (1 or 2), and we use NDCG with these graded judgments for evaluation as detailed below. Furthermore, a recent study [54] showed that even when using only binary relevance judgments for the ASRC dataset, there are considerable performance differences of a wide array of retrieval methods: classical methods [43, 56], feature-based learning-to-rank [4, 18, 55], deep matching models [6, 12, 16, 30] and methods using pre-trained language models [23, 26].

4.2 Learning-To-Rank

We now describe the content-based features, $\mathcal{F}$, used to learn a ranking function $\mathcal{M}$. (See Section 3.2.) Except for the BERT feature, all features were used in the ranking functions in Raifer et al.'s [40] and Goren et al.'s [10] competitions. All the query-document features we use, except for BERT, are representatives from Microsoft's learning-to-rank dataset[8] that can be applied to the plaintext documents in our datasets. Other features are effective document quality measures adopted from work on Web retrieval [2].

Let d and q be a document and a query, respectively, for which a feature vector is defined. The features are: (i) **Okapi**: the BM25 score of d for q, (ii) **LM**: the language-model score of d for q (cf., Equation 5): $-CE(p_q^{MLE}(\cdot) \;||\; p_d^{Dir}(\cdot))$, (iii) **TF**: $\sum_{w \in q} \text{tf}(w \in d)$, (iv) **NormTF**: $\frac{1}{|d|} \sum_{w \in q} \text{tf}(w \in d)$, (v) **LEN**: $|d|$, (vi) **FracStop**: the fraction of term occurrences in d that are stopwords; this feature and the next two were shown to be highly effective document relevance priors [2]; the NLTK stopword list was used (https://www.nltk.org/nltk_data/), (vii) **StopCover**: the fraction of stopwords on the stopword list that appear in d, (viii) **ENT**: the entropy of d's term distribution: $-\sum_w p_d^{MLE}(w) \log p_d^{MLE}(w)$, (ix) **BERT**: The score assigned to d by a (large) BERT model fine tuned for query-based passage ranking over MS MARCO [31]; note that documents in our datasets are short (up to 150 terms). Feature values are min-max normalized. We applied Krovetz stemming and removed stopwords from queries except for inducing the BERT feature.

We follow Raifer et al. [40] and Goren et al. [10] and use the state-of-the-art LambdaMART method [55][9] as the learning-to-rank (LTR) approach. Our ranker consistently outperformed theirs. (Actual numbers are omitted as they convey no additional insight.) Their ranker was a LambdaMART method trained with many content-based features from Microsoft's learning-to-rank dataset. The content-based features we use are a subset of theirs except for BERT. Furthermore, their rankers were trained on ClueWeb09 with TREC's topic titles 1–200, while our LambdaMART was trained with their competitions data [10, 40][10].

4.3 Reference Comparisons

The Okapi, LM and BERT features described above are used as reference comparison methods. They depend only on the current snapshot of the document and the corpus.

The following two reference comparisons are adopted from work on utilizing previous snapshots of the corpus and the document to improve the document representation [1, 8]. We selected representative (highly effective) methods that utilize past snapshots since our approaches also use these snapshots.

The **SepLM** method [8] induces a *separate* language model for three sets of terms. The sets are defined based on how long in the history a term appears in the document's historical snapshots. The language models are mixed and together with a prior are used to score a document in the language modeling framework. The **PastTF** approach [1] adjusts the term frequency (tf) of a term in a document based on its bursts in historical snapshots of the document[11]. The adjusted tf is used in Okapi BM25 or language-model based retrieval yielding the methods **PastTFOkapi** and **PastTFLM**, respectively.

4.4 Evaluation and Training

Since the number of documents ranked for a query per a competition round is at most 8 (see Table 1), we use NDCG@1, NDCG@3 and NDCG@5 as evaluation measures with the graded relevance judgments described above. We use leave-one-out cross validation over query-round pairs to train the LTR models (all based on LambdaMART) and to set free-parameter values of all models. Specifically, we hold out a test query at round i and use for train and validation all other queries at round i. As noted above, we report the average performance over queries and rounds; i.e., over the query-round pairs when the pair was the one held out for testing. Statistically significant performance differences with $p \leq 0.05$ are determined using the two tailed randomization test applied over query-round pairs with 10000 random permutations [45]. Bonferroni correction was applied for multiple-hypothesis testing.

We train the LTR models as follows. For each held out query used for test, we repeat the following process 5 times. We randomly select 3 train queries for validation and all the remaining train queries (27) are used to train the model. Once the values of hyper parameters are set using the validation set, we re-train the model using all train queries including the validation queries. We then apply the five trained models on the held out (test) query and attain five performance numbers (for each evaluation measure) for that query. Their average is the final evaluation score for the query. We use this repeated procedure of splitting the entire training data to train and validation due to the low number of queries. For the unsupervised methods, all train queries were used to set free-parameter values. NDCG@5 served as the optimization goal when training LambdaMART and for setting free-parameter values.

4.5 Free-Parameter Values

The Dirichlet smoothing parameter, μ, used in LM and the mixture model was set to values in $\{50, 100, 200, 300, 500, 700, 800, 900, 1000, 1200, 1500\}$. When LM was used as a feature in LambdaMART, we set $\mu = 1000$ following previous recommendations [56]. The mixture-model mixing coefficients, λ_1 and λ_2, were set to values in $\{0, 0.1, \ldots, 0.9\}$ with the constraint $\lambda_1 + \lambda_2 \leq 1$. The k parameter in the ranking-competition aware model (mixture model) was

[8] www.research.microsoft.com/en-us/projects/mslr

[9] RankLib implementation: https://sourceforge.net/p/lemur/wiki/RankLib/.

[10] A recent study [54] showed that RankSVM is slightly more effective than LambdaMART over the ASRC dataset. Still, we used LambdaMART as it was the ranker in the competitions that resulted in the ASRC dataset [40]: students modified their documents in response to rankings induced by LambdaMART in these competitions.

[11] Two indications for term bursts are used in [1]: one based on content and the other based on the number of document changes in a time interval. Since the latter cannot be used for the ranking competitions data as there are no such intervals, we use only the content-based burst indication.

Table 2: Comparison of the baseline methods. Boldface: best result per column. 'l' and 'o': a statistically significant difference with LM and Okapi, respectively.

	ASRC			Combined		
	NDCG@1	NDCG@3	NDCG@5	NDCG@1	NDCG@3	NDCG@5
LM	.762	.806	.904	.743	**.755**	.824
Okapi	**.766**	**.809**	**.906**	.743	.753	**.825**
BERT	$.664^{l}_{o}$	$.758^{l}_{o}$	$.874^{l}_{o}$	.685	.717	$.791^{l}_{o}$
SepLM	$.706_{o}$	$.769^{l}_{o}$	$.885^{l}_{o}$	$.652^{l}_{o}$	$.710^{l}_{o}$	$.780^{l}_{o}$
PastTFOkapi	$.712^{l}_{o}$	$.776^{l}_{o}$	$.886^{l}_{o}$	**.745**	.752	.815
PastTFLM	$.711_{o}$	.795	.892	$.632^{l}_{o}$	$.691^{l}_{o}$	$.763^{l}_{o}$

set to 3 for TopRank and to 4 for HighImp based on some preliminary experiments with values in $\{1, 2, 3\}$ and $\{2, 3, 4\}$, respectively. k_1 and b in Okapi, when it was used as a method at its own right, were set to values in $\{0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$ and $\{0.3, 0.45, 0.5, 0.55, 0.6, 0.75, 0.9\}$, respectively; when Okapi was used as a feature in LambdaMART, we used the default values: $k_1 = 1.2$ and $b = 0.75$. The number of trees and the number of leaves in a tree in LambdaMART were set to values in $\{250, 500\}$ and $\{2, 3, 5\}$[12], respectively. All other parameters of LambdaMART were set to default values of the implementation (RankLib). The free parameter values for the SepLM baseline [8] were set as follows: $\gamma \in \{0, 0.1, 0.5, 0.9, 1, 1.1, 1.5, 1.6, 1.7, 2.0, 2.3, 2.5\}$; μ_S, μ_M and μ_L were set to values in $\{5, 50, 100, 200, 300, 500, 700, 800, 1000, 1200, 1500\}$; λ_S, λ_M, λ_L ($\lambda_S + \lambda_M + \lambda_L \leq 1$) were set to values in $\{0, 0.1, \ldots, 1\}$. The parameters for the PastTF baselines were set as follows: $\alpha_{burst} \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$; $\alpha_{global}, \beta \in \{-2.0, -1.5, -1.3, -1.2, -1.1, -0.9, -0.5, -0.1, 0.1, 0.5, 0.9, 1, 1.1, 1.2, 1.3, 1.5, 2\}$; $\lambda_1, \lambda_2, \lambda_3$ ($\lambda_1 + \lambda_2 + \lambda_3 = 1$) $\in \{0, 0.1, \ldots, 1\}$; $\mu = 1000$; Okapi's $b_1 = 0.5$ and $k_1 = 1$ following [1].

5 EXPERIMENTAL RESULTS

5.1 Comparison of Baselines

We start by contrasting the performance of the various reference comparison methods we use (i.e., the baselines). Table 2 presents the performance numbers. We see that LM and Okapi outperform the other baselines in almost all relevant comparisons (2 datasets $\times$ 3 evaluation measures); most of the performance differences are statistically significant. Okapi slightly outperforms LM in several cases but not statistically significantly. We note that in a recent study [54], 13 methods including LM, Okapi, feature-based learning-to-rank approaches [4, 18, 55], deep matching models [6, 12, 16, 30] and methods based on pre-trained language models [23, 26] including the BERT baseline we use here, were compared over the ASRC dataset. LM and Okapi where two of the three best performing methods with performance almost identical to that of the best performing method. Furthermore, the rankings induced by LM and Okapi were substantially more robust than those induced by the other 11 methods [54]. Given these findings, and the performance numbers in Table 2, we use LM and Okapi as the two main reference comparisons for our methods in what follows.

We also see in Table 2 that BERT is often outperformed by the other baselines which is in line with previous findings [54]. It was fine tuned on the passage retrieval task in MS MARCO; there is not enough data in the competition datasets to further fine tune it. Furthermore, we found that BERT's performance in the first two rounds of the competitions was substantially better than its performance in later rounds. This finding potentially attests to BERT's sensitivity to ranking incentivized document manipulations; the more the competitions progressed, the more substantial the manipulations were [10, 40]. This finding is also aligned with the relatively low robustness of rankings induced by BERT over the ASRC dataset as recently reported [54].

[12]Higher number of leaves resulted in overfitting.

Table 3: Mixture model (Mix). ASRC+QualityFilter results from ASRC by removing low quality documents. ParsimonLM is the parsimonious language model [15]. Boldface: the best result in a column. 'l', 's' and 'f' mark statistically significant differences with LM, SepLM and PastTFLM, respectively. There are no statistically significant differences between ParsimonLM and Mix-TopRank, Mix-HighImp.

	ASRC			ASRC+QualityFilter	
	NDCG@1	NDCG@3	NDCG@5	NDCG@1	NDCG@3
LM	.762	.806	.904	.770	.817
SepLM	.706	$.769^{l}$	$.885^{l}$	.718	$.776^{l}$
PastTFLM	.711	.795	.892	.719	.801
ParsimonLM	.764	$.811^{s}$	$.906^{s}$	.770	$.820^{s}$
Mix-TopRank	$.773^{s}_{f}$	.797	.903	.777	.805
Mix-HighImp	$.775^{s}_{f}$	$\mathbf{.819}^{ls}$	$\mathbf{.910}^{s}$	$\mathbf{.781}_{f}$	$\mathbf{.829}^{ls}$

As Table 2 shows, the PastTFLM [1] and SepLM [8] methods are consistently — and often statistically significantly — outperformed by LM and Okapi. PastTFLM and SepLM fuse information about past document snapshots at the term and/or language model level. They were designed to improve the representation of the existing document snapshot using its past snapshots, but without an explicit account for potential effects of ranking incentives. We therefore arrive to the conclusion that utilizing historical snapshots of document without accounting for potential ranking incentives can fall short in competitive retrieval settings.

5.2 The Language Modeling Framework

The evaluation of our proposed mixture model is presented in Table 3. We report the performance of using the two suggested ranking-incentives aware models in the mixture model (Section 3.1.1): TopRank (using documents most highly ranked for the query in the past) and HighImp (using documents whose ranking throughout the history has improved the most). The LM[13], SepLM [8] and PastTFLM [1] methods serve as baselines. An additional reference comparison is the parsimonious language model [15], denoted **ParsimonLM**. This is a special case of our mixture model when not using the rank-competition aware generator; i.e., setting $\lambda_1 = 0$ in Equation 3. ParsimonLM was shown to be more effective for Web retrieval than a standard language model [22]. We tune ParsimonLM's free parameters as those of LM and the mixture models.

Table 3 shows the performance for ASRC and ASRC+QualityFilter. The latter is the result of removing from the corpus the documents deemed to be of low quality. (See Section 4 for details.) We do not

[13]Okapi is not used here as the focus is on language modeling; and, its performance was shown above to be statistically indistinguishable from that of LM.

Table 4: The performance of LTR+Agg which adds the history-based features to LTR. LTR+QualityFilter (top table): removing from the LTR ranked list low quality documents. This is equivalent to the LTR method in the bottom table where low-quality documents were removed from the entire corpus. Boldface: best result per column per table. 'l', 'o', 't' and 'q' mark statistically significant differences (per table) with LM, Okapi, LTR and LTR+QualityFilter, respectively.

	ASRC			Combined		
	NDCG@1	NDCG@3	NDCG@5	NDCG@1	NDCG@3	NDCG@5
LM	.762	.806	.904	.743	.755	.824
Okapi	.766	.809	.906	.743	.753	.825
LTR	.800	.826	.916	.772	.794	.845
LTR+QualityFilter	.800	.830	––	.768	.798	.850
LTR+Agg	$\mathbf{.860}^{lo}_{tq}$	$\mathbf{.855}^{lo}_{t}$	$\mathbf{.932}^{lo}_{t}$	$\mathbf{.864}^{lo}_{tq}$	$\mathbf{.847}^{lo}_{tq}$	$\mathbf{.882}^{lo}_{tq}$
	ASRC+QualityFilter			Combined+QualityFilter		
	NDCG@1	NDCG@3	NDCG@5	NDCG@1	NDCG@3	NDCG@5
LM	.770	.817	––	.746	.767	.836
Okapi	.770	.821	––	.751	.767	.837
LTR	.800	.830	––	.768	.798	.850
LTR+Agg	$\mathbf{.860}^{lo}_{t}$	$\mathbf{.859}^{lo}_{t}$	––	$\mathbf{.868}^{lo}_{t}$	$\mathbf{.854}^{lo}_{t}$	$\mathbf{.889}^{lo}_{t}$

use NDCG@5 for ASRC+QualityFilter since after the removal of low quality documents, there were some cases where the retrieved list contained less than 5 (but at least 3) documents. The goal is to study the effectiveness of our mixture model in both a "noisy" setting with low quality documents (ASRC) and a cleaner setting in that respect (ASRC+QualityFilter). These two types of evaluation are conceptually reminiscent of those for Web retrieval with and without spam removal [2]. We do not use the Combined dataset here as it combines documents from two different competitions: Raifer et al. [40] and Goren et al. [10]. Hence, there is no straightforward way to define the historical document sets used by the TopRank and HighImp ranking-competition aware models.

We see in Table 3 that the mixture model with HighImp, Mix-HighImp, is always the best performing method in the table. This finding attests to the merit of analyzing rank promotion patterns over historical corpus snapshots (HighImp) to estimate presumed ranking competition effects. More generally, it attests to the merit in devising relevance estimates that are ranking-incentives aware (**RQ1**; refer back to Section 1). Using previous top-ranked documents (Mix-TopRank) to that end is less effective. We note that while Mix-HighImp does not improve over ParsimonLM in a statistically significant manner[14], it posts more statistically significant improvements over the other methods: Mix-HighImp statistically significantly improves over each of LM and PastTFLM in two relevant comparisons, while ParsimonLM does not improve at all over these methods in a statistically significant manner; Mix-HighImp statistically significant improves over SepLM in 4 relevant comparisons while ParsimonLM does so in 3 relevant comparisons. Among all methods in Tables 2 and 3, Mix-HighImp is the only one that attains statistically significant improvements over LM.

The findings presented above attest to the effectiveness of our mixture model in both noisy (with low quality documents) and cleaner settings. Specifically, using a ranking-incentives aware generator results in the most effective performance in Table 3. Although the performance gains over a standard language model, LM, are not big, our mixture model's performance dominates that of LM as opposed to the other baselines. Furthermore, recall that LM was recently shown to be one of the most effective methods evaluated over the ASRC dataset; specifically, with respect to neural retrieval methods [54].

Finally, we note that to the best of our knowledge, the document language model induced using the mixture model is the first reported in the literature to account for adversarial effects, and more specifically, for those that result from ranking incentives.

5.3 The Learning-To-Rank Framework

Table 4 presents the performance of our LTR+Agg approach (Section 3.2) that adds the Agg features to those used in LTR. The Agg features quantify historical changes of LTR's feature values and the similarities of a document to its past snapshots. In the top table in Table 4 we use LTR+QualityFilter as an additional baseline: removing from the LTR retrieved lists documents defined as of low quality. This approach reflects the potential of using query-independent document quality estimates as in Web retrieval [2]. Hence, LTR+QualityFilter in the top table is equivalent to LTR in the bottom table where low quality documents were filtered out from the entire collection. We do not report NDCG@5 for ASRC when filtering out low quality documents as some of the retrieved lists contain less than five (but at least three) documents. This does not happen for Combined.

We see in Table 4 that our LTR+Agg approach consistently and statistically significantly outperforms LTR. This attests to the effectiveness of the Agg features that quantify historical content changes of the document. More generally, this finding provides further support to the potential merit in devising relevance estimates that are ranking-incentives aware (**RQ1**). The effectiveness of LTR+Agg over corpora from which low quality documents were filtered out (ASRC+QualityFilter and Combined+QualityFilter) attests to the complementary nature of "low quality/spam" and ranking-incentivized content effects (**RQ3**).

Table 4 shows that LTR+Agg also consistently and significantly outperforms all other baselines: LM, Okapi and LTR+QualityFilter. The statistically significant superiority to LTR+QualityFilter in the top table is of special importance since LTR+QualityFilter is based on removal of documents judged by humans to be of low relevance grade and of low quality (**RQ2**).

All the performance numbers reported thusfar were averages over the competitions' rounds. Figure 1 depicts the performance of LTR and LTR+Agg across the rounds of the competitions for ASRC and Combined. The figures for ASRC+QualityFilter and Combined+QualityFilter, and for NDCG@1 and NDCG@3, show the same patterns and are omitted as they convey no additional insight. We see in Figure 1 that the performance of LTR+Agg dominates that of LTR across the rounds.

In summary, LTR+Agg is highly effective on corpora with and without low quality documents (**RQ1, RQ2**). It also outperforms *human filtering* of low-quality documents with low relevance grades

[14] We found that in the third round for ASRC+QualityFilter, Mix-HighImp statistically significantly improved over ParsimonLM. For the other rounds and for ASRC, there were no per-round statistically significant differences.

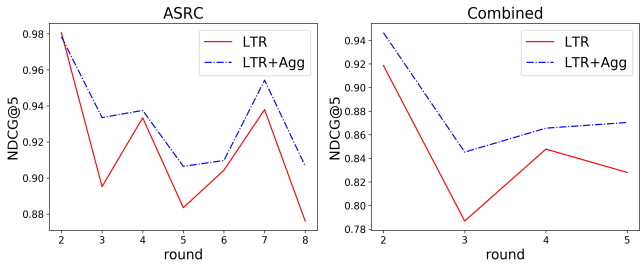

Figure 1: The NDCG@5 performance of LTR and LTR+Agg along the competitions' rounds.

Table 5: The top-10 single-feature rankers. Left and right arrows refer to ASRC and Combined, respectively. '↓' and '↑' indicate positive and negative feature polarity, respectively. Boldface: the best result in a column. 'a': a statistically significant difference with LTR+Agg.

	ASRC			Combined		
	NDCG@1	NDCG@3	NDCG@5	NDCG@1	NDCG@3	NDCG@5
LTR+Agg	**.860**	**.855**	**.932**	**.864**	**.847**	**.882**
↓↓Sim-Min	.822^a	.833^a	.923	.836^a	.837	.870^a
↓↓Sim-Avg	.809^a	.839	.920^a	.849	.824^a	.869^a
↑↑LEN-Std	.819	.830^a	.917^a	.853	.833	.857^a
↑↑NormTF-Std	.805^a	.810^a	.910^a	.857	.823	.857^a
↑↑BERT-Std	.766^a	.808^a	.903^a	.844	.818^a	.864
↓↓Sim-Max	.791^a	.840	.919^a	.779^a	.796^a	.847^a
↓↓LEN	.782^a	.834	.916^a	.740^a	.806^a	.848^a
↓↓LEN-Avg	.804^a	.831^a	.918	.760^a	.798^a	.845^a
↓↓LEN-Min	.818^a	.828^a	.919^a	.780^a	.808^a	.844^a
↓↓ENT-Avg	.799^a	.827^a	.916^a	.816	.801^a	.845^a

(**RQ2**). Furthermore, the performance transcends that of the unsupervised mixture-model-based approach studied above.

Feature Analysis. We performed ablation tests for LTR+Agg by removing one feature at a time. We ordered the 49 features in descending order of the average NDCG@5 relative drop over ASRC and Combined. The ten first features are Sim-Max, BERT-Max, Sim-Min, NormTF-Avg, FracStop-Max, TF, ENT-Min, LEN-Max, Okapi-Std and LEN-Min. Nine out of these ten are Agg features and not the original LTR features (TF is the exception). This finding further attests to the effectiveness of the Agg features.

The performance drop for Sim-Max, the maximal similarity of a current document snapshot with its past snapshots, is statistically significant for all evaluation measures over the two datasets. The only other case of a statistically significant drop was for NDCG@3 of the TF feature over ASRC. Overall, these findings attest to a considerable amount of redundancy between features.

We next use each feature alone as a ranking method. We refer to a feature as of *positive polarity* if ranking by descending order of the values it assigns to documents is superior (in terms of NDCG@5) to ranking by ascending order; if the reverse holds, the feature is of *negative polarity*. We select for the feature the better ranking of the two. Table 5 presents the 10 features whose average NDCG@5 ranking performance over the two datasets is the highest.

Table 5 shows that LTR+Agg is the best performing model. The vast majority of improvements over single features are statistically significant. All features are statistically significantly outperformed by LTR+Agg in at least three out of the six (2 datasets × 3 evaluation measures) relevant comparisons.

We also see in Table 5 that 9 out of the top-10 are Agg features. This finding echoes that in the ablation tests and further attests to the effectiveness of the Agg features. Sim-Min, Sim-Avg and Sim-Max are among these top-10 features and are all of positive polarity; Sim-Max was the top ranked in the ablation tests. These findings mean that high similarity of a document with its past snapshots is an indicator for relevance. Another finding is along the same lines: *all* X-Std features are of negative polarity. That is, high variance in feature values across historical snapshots is an indicator for non-relevance. This finding could potentially be attributed to the fact that significant changes of documents along time might be a signal for increased attempts to improve rankings.

We note that the standard features LM, Okapi and TF, used in LTR and which are not among the top-10 presented in Table 5, are of positive polarity; i.e., surface level similarity of the current document snapshot to the query is a relevance indicator.

6 CONCLUSIONS AND FUTURE WORK

We addressed the ad hoc retrieval task in competitive retrieval settings where document authors might be ranking incentivized; i.e., they might modify their documents' content to improve ranking. We devised content-based relevance estimation methods using two major retrieval frameworks: language modeling and feature-based learning-to-rank (LTR). The methods are not tailored to a specific type of content manipulation strategy.

Empirical evaluation demonstrated the merits of our most effective relevance estimates, specifically, with respect to those which were not designed to address ranking-incentivized content manipulations (**RQ1**, Section 1). We also demonstrated the merits of our learning-to-rank-based approach with respect to using (human created) query-independent document quality estimates (**RQ2**) and their integration (**RQ3**).

Our learning-to-rank (LTR) approach is based on manual feature engineering. Devising representation-based learning approaches that model temporal document modifications is a future direction. Furthermore, the LTR approach, in contrast to the mixture-model-based method, relies on past snapshots of a document. Thus, it is not suitable for cold start settings. A potential future direction to address this issue is to utilize information induced from similar documents with past snapshots.

The rankers we employed do not change along time, although the information they utilize changes based on temporal dynamics. Continuously adapting the rankers to strategic document modifications (cf., work on classification [34]) is left for future work.

Acknowledgments. We thank the reviewers for their comments. The work by Moshe Tennenholtz was supported by funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 740435). The work reported in the paper was also supported in part by the Israel Science Foundation (grant no. 403/22).

REFERENCES

[1] Ablimit Aji, Yu Wang, Eugene Agichtein, and Evgeniy Gabrilovich. 2010. Using the past to score the present: Extending term weighting models through revision history analysis. In *Proceedings of CIKM*. 629–638.

[2] Michael Bendersky, W. Bruce Croft, and Yanlei Diao. 2011. Quality-biased ranking of Web documents. In *Proceedings of WSDM*. 95–104.

[3] Carlos Castillo and Brian D. Davison. 2010. Adversarial Web Search. *Foundations and Trends in Information Retrieval* 4, 5 (2010), 377–486.

[4] Koby Crammer and Yoram Singer. 2001. Pranking with Ranking. In *Proceedings of NIPS*, Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani (Eds.). 641–647.

[5] W. Bruce Croft and John Lafferty (Eds.). 2003. *Language Modeling for Information Retrieval*. Number 13 in Information Retrieval Book Series. Kluwer.

[6] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In *Proceedings of WSDM*. 126–134.

[7] Miles Efron. 2010. Linear time series models for term weighting in information retrieval. *J. Assoc. Inf. Sci. Technol.* 61, 7 (2010), 1299–1312.

[8] Jonathan L. Elsas and Susan T. Dumais. 2010. Leveraging temporal dynamics of document content in relevance ranking. In *Proceedings of WSDM*. 1–10.

[9] Gregory Goren, Oren Kurland, Moshe Tennenholtz, and Fiana Raiber. 2018. Ranking Robustness Under Adversarial Document Manipulations. In *Proceedings of SIGIR*. 395–404.

[10] Gregory Goren, Oren Kurland, Moshe Tennenholtz, and Fiana Raiber. 2020. Ranking-Incentivized Quality Preserving Content Modification. In *Proceedings of SIGIR*. 259–268.

[11] Gregory Goren, Oren Kurland, Moshe Tennenholtz, and Fiana Raiber. 2021. Driving the Herd: Search Engines as Content Influencers. In *Proceedings of CIKM*. 586–595.

[12] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of CIKM*. 55–64.

[13] Zoltán Gyöngyi and Hector Garcia-Molina. 2005. Web Spam Taxonomy. In *Proceedings of AIRWeb 2005*. 39–47.

[14] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan O. Pedersen. 2004. Combating Web Spam with TrustRank. In *Proceedgins of VLDB*. 576–587.

[15] Djoerd Hiemstra, Stephen Robertson, and Hugo Zaragoza. 2004. Parsimonious language models. In *Proceedings of SIGIR*. 178–185.

[16] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of CIKM*. 2333–2338.

[17] Porter Jenkins, Jennifer Zhao, Heath Vinicombe, Anant Subramanian, Arun Prasad, Atillia Dobi, Eileen Li, and Yunsong Guo. 2020. Natural Language Annotations for Search Engine Optimization. In *Proceedings of The Web Conference*. 2856–2862.

[18] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of SIGKDD*. 133–142.

[19] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of SIGIR*. 154–161.

[20] Timothy Jones, Ramesh S. Sankaranarayana, David Hawking, and Nick Craswell. 2009. Nullification test collections for web spam and SEO. In *Proceedings of AIRWeb*. 53–60.

[21] Nattiya Kanhabua, Roi Blanco, and Kjetil Nørvåg. 2015. Temporal Information Retrieval. *Found. Trends Inf. Retr.* 9, 2 (2015), 91–208.

[22] Rianne Kaptein, Rongmei Li, Djoerd Hiemstra, and Jaap Kamps. 2008. Using parsimonious language models on web data. In *Proceedings of SIGIR*. 763–764.

[23] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of SIGIR*. 39–48.

[24] Oren Kurland and Moshe Tennenholtz. 2022. Competitive Search. In *Proceedings of SIGIR*. 2838–2849.

[25] John D. Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR*. 111–119.

[26] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2020. Pretrained Transformers for Text Ranking: BERT and Beyond. *CoRR* abs/2010.06467 (2020).

[27] Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval*. Springer. I–XVII, 1–285 pages.

[28] Ross A. Malaga. 2008. Worst practices in search engine optimization. *Commun. ACM* 51, 12 (2008), 147–150.

[29] David R. H. Miller, Tim Leek, and Richard M. Schwartz. 1999. A hidden Markov model information retrieval system. In *Proceedings of SIGIR*. 214–221.

[30] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *Proceedings of WWW*. 1291–1299.

[31] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019).

[32] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. 2006. Detecting spam web pages through content analysis. In *Proceedings of WWW*. 83–92.

[33] Sérgio Nunes, Cristina Ribeiro, and Gabriel David. 2011. Term Weighting Based on Document Revision History. *JASIST* 62 (12 2011), 2471–2478.

[34] Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. 2020. Performative Prediction. In *Proceedings of ICML 2020*. 7599–7609.

[35] Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of SIGIR*. 275–281.

[36] Kira Radinsky and Paul N. Bennett. 2013. Predicting content change on the web. In *Proceedings of WSDM*. 415–424.

[37] Kira Radinsky, Fernando Diaz, Susan T. Dumais, Milad Shokouhi, Anlei Dong, and Yi Chang. 2013. Temporal web dynamics and its application to information retrieval. In *Proceedings of WSDM*. 781–782.

[38] Fiana Raiber, Kevyn Collins-Thompson, and Oren Kurland. 2013. Shame to be sham: addressing content-based grey hat search engine optimization. In *Proceedings of SIGIR*. 1013–1016.

[39] Fiana Raiber, Oren Kurland, and Moshe Tennenholtz. 2012. Content-based relevance estimation on the web using inter-document similarities. In *Proceedings of CIKM*. 1769–1773.

[40] Nimrod Raifer, Fiana Raiber, Moshe Tennenholtz, and Oren Kurland. 2017. Information Retrieval Meets Game Theory: The Ranking Competition Between Documents' Authors. In *Proceedings of SIGIR*. 465–474.

[41] Nisarg Raval and Manisha Verma. 2020. One word at a time: adversarial attacks on retrieval models. *CoRR* abs/2008.02197 (2020).

[42] Stephen E. Robertson. 1977. The Probability Ranking Principle in IR. *Journal of Documentation* (1977), 294–304. Reprinted in K. Sparck Jones and P. Willett (eds), *Readings in Information Retrieval*, pp. 281–286, 1997.

[43] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of TREC*.

[44] Aécio S. R. Santos, Bruno Pasini, and Juliana Freire. 2016. A First Study on Temporal Dynamics of Topics on the Web. In *Proceedings of WWW*. 849–854.

[45] Mark D. Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of CIKM*. 623–632.

[46] Congzheng Song, Alexander M. Rush, and Vitaly Shmatikov. 2020. Adversarial Semantic Collisions. *CoRR* abs/2011.04743 (2020).

[47] Fei Song and W. Bruce Croft. 1999. A general language model for information retrieval. In *Proceedings of SIGIR*. 279–280.

[48] Junshuai Song, Jiangshan Zhang, Jifeng Zhu, Mengyun Tang, and Yong Yang. 2022. TRAttack: Text Rewriting Attack Against Text Retrieval. In *Proceedings of RepL4NLP@ACL*. 191–203.

[49] Moshe Tennenholtz and Oren Kurland. 2019. Rethinking search engines and recommendation systems: a game theoretic perspective. *Commun. ACM* 62, 12 (2019), 66–75.

[50] Ziv Vasilisky, Moshe Tennenholtz, and Oren Kurland. 2020. Studying Ranking-Incentivized Web Dynamics. In *Proceedings of SIGIR*. 2093–2096.

[51] Yumeng Wang, Lijun Lyu, and Avishek Anand. 2022. BERT Rankers are Brittle: A Study using Adversarial Document Perturbations. In *Proceedings of ICTIR*. 115–120.

[52] Chen Wu, Ruqing Zhang, Jiafeng Guo, Wei Chen, Yixing Fan, Maarten de Rijke, and Xueqi Cheng. 2022. Certified Robustness to Word Substitution Ranking Attack for Neural Ranking Models. In *Proceedings of CIKM*. 2128–2137.

[53] Chen Wu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2022. PRADA: Practical Black-Box Adversarial Attacks against Neural Ranking Models. arXiv:2204.01321

[54] Chen Wu, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2023. Are Neural Ranking Models Robust? *ACM Trans. Inf. Syst.* 41, 2 (2023), 29:1–29:36.

[55] Qiang Wu, Christopher J. C. Burges, Krysta Marie Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval* 13, 3 (2010), 254–270.

[56] Chengxiang Zhai and John D. Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *Proceedings of SIGIR*. 334–342.