# Information Retrieval Meets Game Theory: The Ranking Competition Between Documents' Authors

Nimrod Raifer
Technion, Israel
nimo@campus.technion.ac.il

Fiana Raiber*
Yahoo Research, Israel
fiana@yahoo-inc.com

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

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

## ABSTRACT

In competitive search settings as the Web, there is an ongoing *ranking competition* between document authors (publishers) for certain queries. The goal is to have documents highly ranked, and the means is document manipulation applied in response to rankings. Existing retrieval models, and their theoretical underpinnings (e.g., the probability ranking principle), do not account for post-ranking corpus dynamics driven by this strategic behavior of publishers. However, the dynamics has major effect on retrieval effectiveness since it affects content availability in the corpus. Furthermore, while manipulation strategies observed over the Web were reported in past literature, they were not analyzed as ongoing, and changing, post-ranking response strategies, nor were they connected to the foundations of classical ad hoc retrieval models (e.g., content-based document-query surface level similarities and document relevance priors). We present a novel theoretical and empirical analysis of the strategic behavior of publishers using these foundations. Empirical analysis of controlled ranking competitions that we organized reveals a key strategy of publishers: making their documents (gradually) become similar to documents ranked the highest in previous rankings. Our theoretical analysis of the ranking competition as a *repeated game*, and its *minmax regret equilibrium*, yields a result that supports the merits of this publishing strategy. We further show that it can be predicted with high accuracy, and without explicit knowledge of the ranking function, whether documents will be promoted to the highest rank in our competitions. The prediction utilizes very few features which quantify changes of documents, specifically with respect to those previously ranked the highest.

## KEYWORDS

ad hoc retrieval; game theory; ranking competition

*The paper is based on work done while the author was at the Technion.


*SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan*

DOI: http://dx.doi.org/10.1145/3077136.3080785

## 1 INTRODUCTION

Ad hoc document retrieval models are often based on the assumption of a fixed document corpus — i.e., corpus dynamics is not accounted for. The core challenge is estimating the relevance of a document to the query. The probability ranking principle (PRP) [25] is the theoretical foundation of this practice: to maximize user utility, documents should be ranked by their relevance probabilities.

In practice, document corpora are not static as documents are changed, created or removed. Some of the corpus dynamics, specifically, in competitive search settings (e.g., the Web), results from *ranking incentives* of document authors, henceforth referred to as *publishers*. That is, publishers might modify documents to promote them in rankings induced for queries of interest. These modifications are referred to as search engine optimization (SEO) [16]. Spam filtering, and more generally, using document quality measures as features in learning-to-rank methods [4], are examples of approaches for rank-penalizing documents that have gone through unwarranted modifications (a.k.a., black-hat SEO [16]).

However, existing retrieval approaches, and their theoretical foundations, do not account for future corpus dynamics driven by rankings. For example, it was recently shown that the PRP is sub-optimal in competitive retrieval settings [3] as it can lead to decreased content breadth in the corpus, among other issues.

To estimate post-ranking corpus dynamics, specifically, that caused by responses of publishers to rankings (i.e., document modifications), analysis of the *strategic behavior* of publishers is called for. While types and techniques of SEO strategies were discussed in past work [16], these were not studied as response strategies with respect to rankings induced for specific queries. Rather, they were presented as general actions observed on the Web (e.g., keyword stuffing and content copying).

Furthermore, there are no studies, to the best of our knowledge, that analyze publishers' strategies with respect to retrieval models and their foundations; namely, the effect, over time, on features used for ranking. Such analysis is important for incorporating strategy predictions (estimates) in, and addressing their effects on, retrieval approaches. A case in point, it was shown that if the actual writing quality of publishers for topics is known, then this information can be used in non-deterministic retrieval models to promote content breadth in the corpus, and therefore improve search effectiveness along time [3]. More generally, analysis of the strategic behavior of publishers is crucial for setting theoretical foundations for handling post-ranking corpus dynamics. The same way user modeling is

important for interactive information retrieval models [30], modeling the strategic behavior of publishers in response to induced rankings is important for addressing post-ranking corpus dynamics in retrieval models.

We present a novel initial theoretical and empirical analysis of the (temporal) strategic behavior of publishers in terms of changes they introduce to documents in response to induced rankings. The analysis is performed in the context of classical ad hoc retrieval models in two respects. First, we focus on content-based retrieval and accordingly content manipulation. Analyzing post-ranking strategies of changing hypertext, hyperlinks and affecting clicks or any additional signal that can be used for relevance estimation is outside the scope of this paper. Nevertheless, we note that (i) content-based relevance estimates (e.g., Okapi BM25 and language-model-based estimates) are among the most important ones used in learning-to-rank approaches applied over Web data [20]; (ii) content manipulation techniques are quite pervasive, specifically, over the Web [16]; and, (iii) for experiments we use a state-of-the-art learning-to-rank approach applied with content-based estimates. Second, we empirically study content manipulation in terms of the building blocks of classical, content-based, retrieval methods. These include document-query surface-level similarities [14] and query-independent document relevance priors [4].

Performing empirical analysis of the "ranking competition" between publishers whose incentive is to have their documents ranked high, even if assuming the availability of a large-scale log of a search engine, is a major challenge due to the numerous dynamic aspects that affect this competition. Over the Web, pages emerge and disappear, the search engine's index coverage changes rapidly, the ranking function, as well as estimates it utilizes, change throughout time and across sets of users and queries. Furthermore, different publishers cannot necessarily employ the same document modifications, and many modifications are not content-based as the ranking function also considers non content-based relevance signals.

Given that our goal, as described above, is to study the strategic behavior of publishers in the scope of the foundations of classical content-based retrieval models, we performed controlled empirical analysis by organizing ranking competitions between students in a course. Two basic conditions were set in these competitions. First, the students were not aware of the ranking function, nor of the actual features it used. Second, the students were incentivized to write quality documents that would be ranked high by the ranking function. As shown below, the dataset allowed to gain interesting and important observations about potential strategic behavior of publishers in a ranking competition.

An important observation that emerged in the competition analysis that we present is that publishers were gradually making their documents become more similar, in several respects, to those most highly ranked in previous rankings[1]. An interesting fundamental question that follows is whether this competing strategy can be theoretically justified given the information available to publishers: observations of past rankings and little to no knowledge of the ranking function. To address this question, we present a novel game theoretic analysis of the ranking competition as a *repeated game* [1]. Our main theoretical result with respect to the minmax regret equilibrium of the game [17] provides formal support to the merits of this publishing (competing) strategy.

In addition to analyzing the ranking competition theoretically and empirically, we set as a goal to predict whether a document would be ranked the highest given that this was not the case in the previous ranking; the predictor does not have explicit knowledge of the ranking function. Interestingly, relying on very few features that quantify the extent to which the document was changed and became similar to a document previously ranked the highest can yield high accuracy prediction. These features are inspired by the *cluster hypothesis* [18], and more specifically, one of the important operational premises that it gave rise to: "similar documents should receive similar retrieval scores" [9]. Thus, in lack of knowledge of the ranking function, the predictor essentially uses inter-document similarities as proxies for retrieval score similarities.

Our contributions can be summarized as follows.

- We present the first dataset of query-based ranking competitions between publishers. The focus is on content manipulation.
- We present an empirical analysis of publishers' strategies employed in the competitions.
- We present a novel game theoretic analysis of the ranking competition as a repeated game. The main result of analyzing the minmax equilibrium of the game provides formal support to the merits of a key strategy employed by publishers in our games.
- We show that, in our setting, it is possible to predict with high accuracy whether a document will be promoted to the highest rank in the next ranking. The prediction is based on very few features and does not rely on explicit knowledge of the ranking function.

## 2 RELATED WORK

There is much work on identifying, characterizing and addressing unwarranted (a.k.a. black-hat SEO [16]) actions of publishers [6]. In contrast, we focus on the strategic behavior of publishers when applying legitimate content-based manipulations.

Studies of the dynamic aspects of interactive retrieval focus on changes of queries and the ranking function (e.g., [15, 27, 30, 31]). Changes of clickthrough patterns were also studied [27]. The dynamics of the collection as a result of the ranking competition, which is our focus, was not addressed.

There has been work on studying and predicting the dynamics of the Web collection (e.g., [23, 26]), where the main operational goals were improving crawling policies and personalizing content delivery. Past versions of a Web page were used to improve its representation for ranking [13]. However, in contrast to our work, the dynamics has not been studied with respect to the ranking competition between publishers.

Recently, the publishers' ranking competition was analyzed using a game theoretic approach [3]. In contrast to our work, the assumption was that publishers have full knowledge of the ranking function, the competition was not analyzed as a repeated game, and no empirical analysis was presented.

[1]This strategy is conceptually reminiscent of the black-hat weaving and stitching content-based SEO techniques applied over the Web [6, 16] where content from legitimate pages is copied to spam pages so as to promote them.

The merits of non-deterministic ranking functions from [3] were argued using a *simulation* of a ranking competition between publishers who stuff query terms in documents [2]. In contrast with our work, publishers were assumed to know the basic (Okapi BM25) ranking function, there was no theoretical analysis of the competition and no analysis of non-simulated (real) ranking competitions.

A game theoretic approach was used to devise query-based ranking mechanisms that (i) maximize social welfare for ambiguous queries, by diversifying search results that are assumed to be scanned using random sequential search [12]; and (ii) balance relevance and monetization [11]. In contrast to our work, the competition between documents' authors (publishers) was not studied.

Game theoretical analysis has also been applied for adversarial classification [8, 10] and for optimizing learning-to-rank functions in non-adversarial retrieval settings [28]. We address the competitive (adversarial) ad hoc retrieval setting using different theoretical and empirical analyses.

## 3 GAME THEORETIC ANALYSIS

We analyze the ranking competition as a *repeated game* [1]. Analyzing the minmax regret equilibrium of the game yields a formal result that helps to explain a key strategy employed by publishers in the ranking competitions we organized as described in Section 5.

In what follows we assume that a query $q$, and some document ranking function (details below), have been fixed. Let $N = \{1, 2, \ldots, n\}$ be a set of $n$ publishers (documents' authors) that would like to have their documents ranked high for $q$. Let $D_i$ be a finite set of documents that publisher $i$ can (or might) write to convey the information she wants to share. For ease of presentation, and to avoid technical tie-breaking issues, assume $D_i \cap D_j = \emptyset$ for any $i, j \in N, i \neq j$. Let $D = \cup_{i=1}^{n} D_i$ be the set of all documents that can be written by the publishers.

We assume a complete linear ordering over $D$, denoted $<$. Such ordering can be based, for example, on a single (numeric) feature in a document representation[2]. Alternatively, the distance, under some representation, to a document which serves as a reference point (e.g., a document ranked the highest at some point) can serve to induce the ordering. Thus, for ease of exposition we can associate $D$ with elements in the interval $[0, 1]$. A document ranking function for $q$ is a mapping $r : D \rightarrow \Re_+$. For simplicity (and avoiding tie-breaking), we assume $r(d_i) \neq r(d_j)$ for any $d_i, d_j \in D$.

*Definition 3.1.* $RSP(D_1, \ldots, D_n) = RSP(D)$ denotes the single peak ranking functions. These are functions $r$ defined over $D$, such that for no $d \in D$, there are $d_i, d_j \in D$, $d_i < d < d_j$ such that $r(d_i) > r(d)$ and $r(d_j) > r(d)$.

For example, linear learning-to-rank functions [20] are single peak with respect to each feature. The negative KL divergence used in the language modeling framework [19] is a single peak function over the multinomial distributions in the simplex by the virtue of being a concave function. However, the most effective ranking functions (e.g., those utilizing non-linear learning-to-rank methods) are not single peak. Nevertheless, it is important to keep in mind that we analyze the dynamics from the point of view of publishers who have no information about the ranking function except for that inferred by observing induced rankings. That is, publishers may assume, and act based on the belief, that the ranking function is single peak. Indeed, as shown in Section 5, the participants (publishers) in our ranking competitions can be viewed as searching for the structure of a single-peak ranking function for various features, although the ranking function is not single-peak.

Below we care only about the relative ranking of documents in $D$; thereby, we consider the possible total ordering induced by the ranking function over $D$; there are finitely many such orderings. With a slight abuse of notation we will therefore refer to $RSP(D)$ as the set of possible single-peak orderings of documents in $D$.

Let $D^0 = \{d_1^0, \ldots, d_n^0\}$ be an initial set of documents where $d_i^0 \in D_i$ is the initial document published by publisher $i$. We assume that each $i \in N$ possess no information at the beginning about the function $r \in RSP(D)$, beyond knowing it is a single-peak ordering. Consider $t$ rounds, $l = 1, 2, \ldots, t$, in each of which each player $i$ chooses a document $d_i \in D_i$, and obtains a utility of 1 if $d_i$ is ranked first and 0 otherwise. Herein, a publisher or her document is called "winner" if the document was the highest ranked; all other publishers and their documents are called "losers". Let $TO(D)$ be the set of possible total ordering over $D$. Notice that selecting $d_i \in D_i$ for every $i \in N$ determines an ordering over the selected documents by the single-peak function $r \in RSP(D)$. The *strategy* of $i$ at round $l$ is defined as a function from the history of previously selected actions and outcomes (i.e., orderings) of all publishers, to the document selected by publisher $i$. The outcome at each round can be associated with a subset $R \subseteq RSP(D)$ of the possible single peak functions, as it rules out particular orderings. Henceforth, $R$ is referred to as the *knowledge state*, as it captures the set of currently possible single peak orderings based on the observations received.

The *publishers ranking game* just described is a *repeated game* [1]. In a repeated game, the same game is repeatedly played in rounds (iterations). Specifically, at each round a publisher publishes a document, but a strategy in each round may relate to all information observed so far; e.g., the documents published and rankings induced in previous iterations. Accordingly, given the initial document set $D^0$, and the total number of rounds $t$, the set of possible strategies for player $i$ is denoted $S_i(t, d_i^0)$.[3] The utility $U_i(t, d_i^0, s_1, \ldots, s_n)$, where $s_j \in S_j(t, d_j^0)$ for every $j \in N$, is the sum of utilities of player $i$ in rounds $1, 2, \ldots, t$ given the corresponding strategies.

We now introduce a slight modification to the utility obtained by player $i$ in a round to capture the cost of modifying documents. This cost reflects both the actual effort involved in changing a document and the "penalty" incurred by potentially drifting from the actual document $i$ planned to publish. Assume there is some negligible cost $C$, i.e., $C|D| < 1$, where $eC > 0$ is the cost for changing document $d$ to document $d'$ in distance $e$ (assume standard distance on $[0, 1]$) in a single round. Formally, the utility of publisher $i$ in round $l$ will be based on its ranking (either first or not) minus the cost of changing the document written in round $l - 1$.

Given the game described above, a major challenge is to define an appropriate solution concept which predicts behavior in the game. The classical solution concept in game theory is the celebrated Nash

[2] In this case, the analysis below applies to each feature in a document representation assuming that the values of others were fixed.

[3] $S_i(t, d^0)$ encodes all possible documents published by $i$ at any round of the game given the previous potential orderings.

equilibrium, which is a strategy profile, one for each player, for which unilateral deviations are not beneficial (i.e., any single player cannot gain by deviating from her strategy assuming the others stick to their strategies). This solution concept always exists in finite games with complete information if players are allowed to use mixed strategies, and has been also extended to games with incomplete information where there are Bayesian assumptions about the actual game being played. However, our setting does not exhibit such stylized assumptions, and we need to appeal to other solution concepts. In particular, in a minmax regret equilibrium [17], we consider strategy profiles such that each player (publisher) minimizes her regret when compared to the best response she could have played had she known the exact environment state (e.g., the exact ranking function) assuming others stick to their strategies; and this holds for all players simultaneously.

Given a strategy profile $s = (s_1, \ldots, s_n)$ the regret of $i$ is $max_x U_i(t, d_i^0, x, s_{-i}) - U_i(t, d_i^0, s)$; $s_{-i}$ denotes the strategy profile applied by all players except for $i$. A strategy profile $s = (s_1, \ldots, s_n)$ is *minmax regret equilibrium* if for every $i$, $s_i$ minimizes regret given $s_{-i}$ [17]. Given the defined publishers game we can now show that:

Theorem 3.2. *Any publishers game has a minmax regret equilibrium.*

Proof. We construct the following equilibrium. Let $R$ be the knowledge state at the beginning of a given round $l$. At the beginning of round 1 all ranking functions in $RSP(D)$ are possible, while at each following round the knowledge state can only shrink in terms of the number of ranking functions it contains. Let $V_l \subseteq [0, 1]$ be the set of documents which correspond to possible peaks of the functions in the knowledge state $R$; let $d_i^{l-1}$ be the most recent document published by $i$. Let $v_i^l \in D_i \cap V_l$ such that $|v_i^l - d_i^{l-1}|$ is minimal; if two documents have this minimal distance one is arbitrarily selected. The document published by $i$ in round $l$ would be $v_i^l$. (In the first round it is $d_i^0$.) We now prove that this strategy of $i$ minimizes its regret.

Let $V_t$ be the knowledge state at the beginning of the last round $t$; $V_t$ may result from arbitrary publishers' behavior in rounds $1, 2, \ldots, t-1$. No publisher $j \neq i$ will publish a document not in $V_t$ as otherwise she cannot win (i.e., this strategy would be dominated). Hence, $i$'s publishing a document out of $V_t$ is dominated by publishing the previous document. (This has no cost, and publishing out of $V_t$ cannot result in a win.) On the other hand, since any $v \in V_t$ can be a winner, the worst regret would be for not publishing $v_i^t$ as defined above. This is because $v_i^t$ might be the winner from this point on, by the virtue of being in $V_t$, but it incurs minimal cost. Thus, minimizing regret in the last round is achieved by selecting $v_i^t$ as prescribed. By induction, using the argument from above results in $i$'s strategy minimizing regret in every round. □

Two corollaries follow the proof:

Corollary 3.3. *The above constructed equilibrium is also a subgame perfect equilibrium.*

Namely, if an arbitrary sequence of documents has been selected up to round $l < t$, then in the remaining game, given the information provided so far on the potential peaks, following each player's strategy in the remaining rounds is still a minmax regret equilibrium.

Corollary 3.4. *Losers at round $l-1$ will publish in round $l$ documents that become closer to (i.e., more similar) to that of the winner from round $l-1$.*

Proof. Assume wlog that a publisher who lost round $l-1$ published $d_j \in [0, 1]$ that satisfies $d_j < d_w$ where $d_w \in [0, 1]$ was the winning document. As selecting any $d_{j'} < d_j$ is dominated given the knowledge state gathered, and the regret for publishing $d_{j'''} > d_w$ is higher than that of publishing $d_{j''}$ such that $d_w > d_{j''} > d_j$, we get the corresponding phenomenon. Notice that this will also imply that current winners will not change their documents. □

Thus, Corollary 3.4 helps to explain a key strategy employed by publishers in the competitions we organized as we show below; namely, mimicking the winners.

## 4 DATA

As discussed in Section 1, our goal is to analyze content-based ranking competitions so as to shed light on the strategic behavior of publishers. Since there are no publicly available datasets that can be used to that end, we organized repeated ranking competitions. The resulting dataset is available at https://github.com/asrcdataset/asrc. (See details in Appendix A.) We next describe the essentials of the competition.

Fifty two senior-undergrad and grad students in an information retrieval course were the publishers. The competition included 31 different repeated matches, each of which was with respect to a different TREC's ClueWeb09 query. Each student participated in three matches. Five students competed against each other in all matches except for one in which six students competed.

The competition was run for eight rounds; i.e., there were eight matches per query. Before the first round, an example of a relevant document was provided for each match (query). Students were incentivized by course-grade rewards to edit their documents along the rounds so as to have them ranked as high as possible.[4] As from the second round, students participating in a match were presented with the ranking of documents submitted in the previous round by all competitors in the same match.

All documents were unstructured plain text of up to 150 terms. The document ranking model was based on the state-of-the-art LambdaMART [29] learning-to-rank approach integrating three classes of features. The first are query-dependent features, such as **QueryTermsRatio** (ratio of query terms appearing in a document) and **LMIR.DIR** (language-model-based similarity of a document to the query). The second class of features are query-independent document quality measures [4, 21], including **Entropy** (entropy of the term distribution in a document) and **StopwordsRatio** (stopwords to non-stopwords ratio in a document). Increased entropy and occurrence of stopwords attest to content breadth and hence to high prior probability of relevance [4].

The feature in the third class, **SimInit**, was used to incentivize students to write documents that drift from the initial relevant document shared by all students competing in the same match: it is

[4] Students were assigned with unique IDs and all data was anonymized.

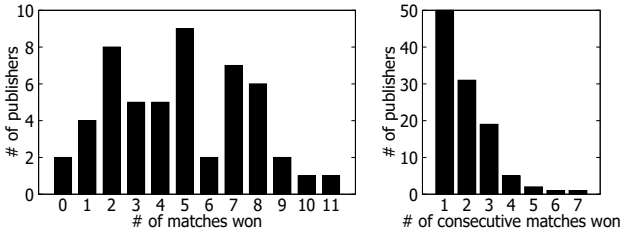


**Figure 1: The number of (consecutive) matches won (x-axis) by a given number of publishers (y-axis).**

based on the language-model similarity of a document to the initial relevant document. We note that in practical scenarios, publishers would rarely change their documents so they will not include the information originally intended for sharing. Indeed, in the theoretical analysis presented in Section 3, a cost was assigned to changes of documents. Yet, as we show below, the conclusions we draw about strategies are aligned with our theoretical results.

Documents (manually) classified as keyword stuffed were penalized in the ranking. Information about the ranking function and the features it utilizes was not disclosed to students. The resulting collection contains (i) 1279 documents: 31 initial relevant documents and 1248 documents created by students, 897 of which are unique[5]; (ii) keyword stuffing annotations; and (iii) exhaustive relevance judgments. Appendix A provides additional details of the collection and ranking model.

## 5 EMPIRICAL ANALYSIS

In the following analysis, winner (loser) is a document (or publisher thereof) which was (not) ranked first in a match.

### 5.1 Analysis of wins

Figure 1 (left) presents the number of matches won by a given number of publishers (students). The competition included 248 distinct matches (8 rounds × 31 matches per round). Each student was assigned with exactly 3 queries; hence, the maximum number of matches a student could win is 24. We see that only two of the students did not win even a single match, attesting to the students' engagement in the competition. The maximum number of matches won was 11, less than half of the maximal possible number of wins, indicating that the competition was dynamic.

Figure 1 (right) presents the number of consecutive matches won by a given number of students; the maximum is the number of rounds (eight). We see that most students could retain the first rank for at most three rounds. Only a small number of students retained the first rank in more than four rounds. This finding further attests to the strong competition held between the students.

### 5.2 Analysis of strategies

By Corollary 3.4, to win matches, losers in previous rounds will publish documents that become similar to that of the winner from the preceding round. Accordingly, we next analyze the similarity of documents that did not win a match (losers) to the winner over a series of rounds in which these documents remained losers. The similarity to the winner is estimated with respect to some of the features used to rank documents which were presented in Section 4.

The QueryTermsRatio and LMIR.DIR features quantify the query-document match; LMIR.DIR is a representative query-document surface-level similarity estimate [14]. The Entropy and StopwordsRatio features are among the most effective query independent content-based document relevance priors reported in the literature [4]. Hence, the analysis of the strategic behavior of publishers we present next relies on estimates that constitute the foundations of classical content-based ad hoc retrieval approaches.

Figure 2 depicts the average values of the features for documents that were losers in at least four consecutive rounds before winning a match[6]. We distinguish between documents whose feature value four rounds before winning a match was lower than or equal to that of the winner (**L**≤**W**) and those whose feature value was higher than that of the winner (**L**>**W**). We also present the average feature values of winners (**W**).

We see in Figure 2 that the average feature values of winners remain relatively stable along the competition; thus, winner documents, often written by different publishers, tend to be quite similar along a few dimensions (features).

Figure 2 also shows that, in general, Entropy often decreases along rounds and QueryTermsRatio increases. This attests to high content repetition in winner documents that might result from high occurrence of query terms. SimInit decreases which is potentially due to our rewarding diversification with respect to the initial relevant document.

More generally, we observe a clear trend throughout the competition: feature values of loser documents which became winners were becoming closer, often gradually converging, to those of winners from previous rounds regardless of their initial values. That is, in lack of knowledge of features used for document ranking, losers were mimicking winners and thereby indirectly affecting these features. This finding is in accordance with Corollary 3.4.

## 6 PREDICTING WINNERS

Given that loser publishers apply the strategy of mimicking the winners, an interesting challenge rises: leveraging aspects of this strategy to predict, without using explicit knowledge of the ranking function, which loser publisher in round $l-1$ will win round $l$ assuming that a previous loser indeed wins this round.[7]

For prediction, we represent each document as a feature vector and define two sets of features (details below) that quantify the extent to which the document becomes more similar to the winner of the previous round. The features in the first set are estimates of this similarity on a *macro* level, where documents are treated as

[5]Several students submitted the same document over a few rounds; e.g., if the document was the highest ranked in a previous round.

[6]Similar trends were observed for other features used by the ranking model and for losers that lost in at least three or five consecutive rounds. These results are omitted as they convey no further insight.

[7]Predicting which publisher will win round $l$ *regardless* if it won round $l-1$ is a challenge for future work. As stated in the proof of Corollary 3.4, and as observed in the competitions, winners did not tend to change their documents. This is a fundamental difference with the dynamics of loser documents which makes this prediction task challenging. For example, many of the dynamics-based features defined below for predicting whether a loser will turn to a winner are degenerated for winner documents which do not change.

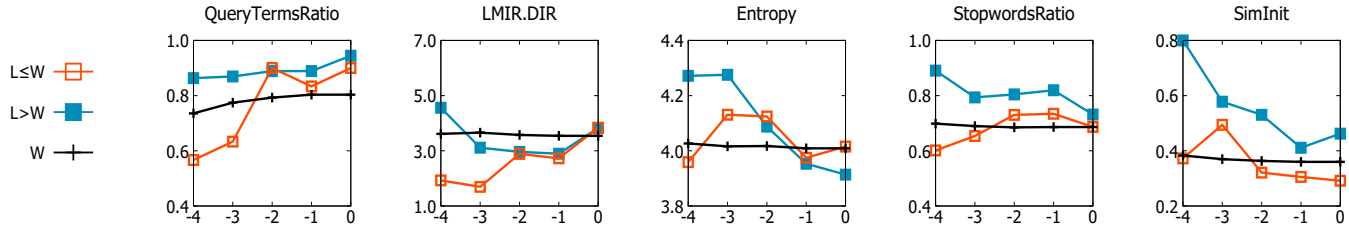


**Figure 2: Averaged feature values of documents that were losers in at least four consecutive rounds before becoming winners, and whose feature values four rounds before winning were either lower or equal ('L≤W') or higher ('L>W') than those of the winner. 'W': averaged feature value of the corresponding winners. x-axis: (minus) number of rounds before a document won a match. The values of LMIR.DIR are scaled by** 100**.**

whole units. The features in the second set are *micro* level similarity estimates that allow to analyze the potential actions taken by publishers to make their documents similar to the winner.

## 6.1 Features

The features in the first set, henceforth **Macro** features, are estimates of the bag-of-terms textual similarities (denoted SIM) between the document in round $l$ (**D**), the document written by the same publisher in the previous round $l-1$ (**PD**) and the winner of the previous round $l-1$ (**PW**). The Cosine between tf.idf vector representations of documents is the similarity estimate. Three estimates are used: **SIM(D,PD)**, **SIM(D,PW)** and **SIM(PD,PW)**.

Using these inter-document similarity measures is inspired by the *cluster hypothesis* [18] which states that "closely associated documents tend to be relevant to the same requests". More specifically, an important operational manifestation of the cluster hypothesis is the premise that effective retrieval methods should assign similar documents with similar retrieval scores [9]. Based on the premise, given that the predictor we devise has no explicit knowledge of the ranking method used, inter-document similarities can potentially serve as proxies for similarities between retrieval scores.

The features in the second set, henceforth **Micro** features, focus on potential actions of publishers to make their documents similar to PW, the winner of the previous round. A document becomes similar to the winner, based on a bag-of-terms representation, if terms from the winner are added and terms not in the winner are removed. Accordingly, given a set $\mathcal{S}$ of terms, **ADD(PW)** and **RMV(PW)** are the number of unique terms $t \in \mathcal{S}$ used in PW that were added to, or removed from, the document, respectively. Similarly, **ADD(¬PW)** and **RMV(¬PW)** are the number of unique terms $t \in \mathcal{S}$ *not* used in PW that were added to or removed from the document, respectively. We define three term sets $\mathcal{S}$: (i) query terms (**Query**), (ii) frequent terms, specifically, stopwords (**Stopwords**), and (iii) non-frequent terms not in the query (**¬Query¬Stopwords**).[8]

Overall, we use 15 features: 3 Macro (SIM(D,PD), SIM(D,PW), SIM(PD,PW)) and 12 Micro ({ADD(PW), RMV(PW), ADD(¬PW), RMV(¬PW) } × {Query, Stopwords, ¬Query¬Stopwords}).

The Macro features, which quantify temporal inter-document similarity changes, are ranking-model agnostic. The Micro features are based on temporal changes of addition/deletion of terms. While term-based information (e.g., query-terms occurrence) would be used by any reasonable ranker, the prediction model uses no explicit knowledge about how this information is used by the non-linear ranker applied in the competition, nor about other features used for ranking.

## 6.2 Prediction setup

In each round of the competition, queries for which the winner of the previous round remained the winner were discarded, as our goal is to predict which *loser* publisher in a previous round will win the current round. Thus, the number of queries considered in each round ranges from 6 to 26 (out of all 31 queries).

We used the features[9] from Section 6.1 for binary classification with logistic regression (**LReg**), linear SVM (**LSVM**), polynomial SVM (**PSVM**) and random forests (**RForest**) via the scikit-learn library [22]; the two classes are winner and loser. To train the classifiers and set hyper-parameter values, we used leave-one-out cross validation over rounds. The documents submitted by students with respect to all considered queries in a round served for testing; those submitted in the remaining six rounds, excluding the first, served for training. Prediction was performed per query: the document in the current round which was written by a loser publisher from the previous round and which was assigned the highest classification score was predicted the winner; all other documents were predicted to be losers.

Prediction effectiveness is measured using **Accuracy**: the percentage of documents correctly predicted as winners or losers, and **F1**: harmonic mean of Precision and Recall.[10] Values are averaged over queries and test folds. Statistically significant effectiveness differences are determined using the two-tailed paired t-test with $p \leq 0.05$ applied over queries.

The hyper-parameter values of the classifiers were selected to optimize Accuracy over the train set. For LReg, LSVM and PSVM, the value of the regularization parameter is in $\{1, 10, 50, 100\}$.[11] The degree of the polynomial SVM (PSVM) was in $\{2, 3, 4, 5\}$. The number of trees and leaves for RForest were selected from {10, 50,

[8] A term is considered a stopword if it is among the 100 most frequent alphanumeric terms in the ClueWeb09 Category B corpus.

[9] Feature values were min-max normalized per query.

[10] Precision is the fraction of correctly predicted winners out of all documents predicted to be winners. Recall is the fraction of winners correctly predicted as winners.

[11] LReg, LSVM and PSVM were trained with L1 regularization.

**Table 1: Prediction effectiveness of the four classifiers (LReg, LSVM, PSVM, RForest) and the baselines. The performance differences between each of the classifiers and each of the baselines are statistically significant. All the differences with RForest are statistically significant. Bold: the best result in a row. Note: F1 of AllLosers is 0 due to zero Recall.**

| | Random | Majority | AllWinners | AllLosers | LReg | LSVM | PSVM | RForest |
|---|---|---|---|---|---|---|---|---|
| Accuracy | 0.627 | 0.685 | 0.247 | 0.753 | 0.849 | 0.859 | 0.867 | **0.878** |
| F1 | 0.242 | 0.363 | 0.396 | 0.000 | 0.695 | 0.712 | 0.730 | **0.752** |

100, 500} and {10, 20, 30}, respectively. All other hyper-parameters were set to their default values [22].

## 6.3 Prediction effectiveness

*Main result.* We compare the prediction effectiveness of the aforementioned classifiers with that of four baselines. All prediction algorithms predict as winner(s) documents whose publishers lost the previous round. (i) **Random**: a single winner is randomly selected; (ii) **Majority**: the document whose publisher won the majority of past rounds for the query is predicted the winner (ties are broken arbitrarily); (iii) **AllWinners**: all documents are predicted winners, in which case only one document per query is correctly predicted; and, (iv) **AllLosers**: all documents are predicted losers, in which case all but one of the documents are correctly predicted as losers. The results are presented in Table 1. Although the four classifiers (LReg, LSVM, PSVM and RForest) utilize no knowledge of the ranking model, they predict with high effectiveness the winner of the current round. Moreover, the differences in prediction effectiveness between each of the classifiers and each of the four baselines are substantial and statistically significant. These findings attest to the ability to predict winners from previous losers in our competitions based on macro-level and micro-level manipulation strategies of publishers.

Among the four classifiers, the lowest performance is posted by LReg, while the highest is posted by RForest. Hence, in the analysis to follow we focus on RForest.

*Feature analysis.* We next study the relative effectiveness of the *sets* of features used in RForest. Recall that the 15 features belong to two sets: Macro and Micro. The Micro features belong to three subsets: Query, Stopwords and ¬Query¬Stopwords. In Table 2 we compare the prediction effectiveness of training RForest using different combinations of these (sub)sets of features. We present for reference the effectiveness of the Majority, AllWinners and AllLosers baselines. We see that using even a single (sub)set of features yields prediction effectiveness that statistically significantly surpasses that of the baselines. Among the three subsets of Micro features, the query-term-based features (Query) are the most effective. Integrating all three subsets leads to prediction effectiveness that always statistically significantly surpasses that of using either one or two of the subsets. We also see that using Micro features alone leads to slightly higher effectiveness than using only Macro features; the difference is not statistically significant. Yet, combining both sets yields the highest prediction effectiveness. These findings suggest that the Micro and Macro features, as well as the three subsets of Micro features, are complementary to some extent.

**Table 2: Using subsets of features for prediction. All differences with respect to Majority, AllWinners, AllLosers and Macro+Micro are statistically significant. Bold: best result in a column.**

| | Accuracy | F1 |
|---|---|---|
| Majority | 0.685 | 0.363 |
| AllWinners | 0.247 | 0.396 |
| AllLosers | 0.753 | 0.000 |
| Query | 0.821 | 0.635 |
| Stopwords | 0.809 | 0.594 |
| ¬Query¬Stopwords | 0.796 | 0.587 |
| Query+Stopwords | 0.826 | 0.650 |
| Query+¬Query¬Stopwords | 0.825 | 0.648 |
| Stopwords+¬Query¬Stopwords | 0.813 | 0.617 |
| Micro = Query+ Stopwords+¬Query¬Stopwords | 0.837 | 0.673 |
| Macro | 0.836 | 0.671 |
| Macro+Query | 0.851 | 0.702 |
| Macro+Stopwords | 0.849 | 0.694 |
| Macro+¬Query¬Stopwords | 0.847 | 0.692 |
| Macro+Micro (all features) | **0.878** | **0.752** |

We next study the effectiveness of *individual* features. Table 3 presents the Accuracy of ablation tests performed upon RForest.[12] We also report Δ**MRR**: the mean difference between the reciprocal ranks of the *actual winner* when documents are ranked in descending and ascending order of individual feature values. We first see that removing any single feature statistically significantly hurts Accuracy. This attests to the complementary nature of the features.

The negative ΔMRR of SIM(D,PD) indicates, as expected, that to win a match, a loser publisher should change her document with respect to the previous round. The positive ΔMRR of SIM(PD,PW) and SIM(D,PW) suggest that the document should be similar to the winner (from the previous round) in the previous and current rounds so as to win the match. This finding is aligned with Corollary 3.4.

The ΔMRR of features in the Query and Stopwords subsets indicate that adding (removing) query terms is always good (bad) practice for becoming the winner, regardless of whether these terms were used by the winner. This finding is further supported by the observations about QueryTermsRatio in Section 5.2. In contrast, removing (adding) frequent terms, i.e., stopwords, is always good (bad) practice, regardless of the use of stopwords by the winner. The ΔMRR of features in the ¬Query¬Stopwords subset, which refers to terms that are neither query terms nor stopwords, imply that to

[12]Similar patterns were observed for F1. These results are omitted as they convey no additional insight.

**Table 3: Ablation tests: Accuracy of RForest when trained without one feature. RForest's Accuracy with all features is 0.878. All differences with RForest are statistically significant. ΔMRR: the mean reciprocal ranks difference of the winner when ranking documents in descending and ascending order of feature values.**

| Macro Features | | | Micro Features | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Query | | Stopwords | | ¬Query ¬Stopwords | |
| Feature | Ablation | ΔMRR | Feature | Ablation | ΔMRR | Ablation | ΔMRR | Ablation | ΔMRR |
| SIM(D,PD) | 0.829 | −0.136 | ADD(PW) | 0.844 | 0.130 | 0.840 | −0.219 | 0.841 | 0.183 |
| SIM(D,PW) | 0.837 | 0.168 | RMV(PW) | 0.851 | −0.043 | 0.843 | 0.043 | 0.857 | −0.081 |
| SIM(PD,PW) | 0.820 | 0.184 | ADD(¬PW) | 0.840 | 0.104 | 0.856 | −0.023 | 0.849 | −0.053 |
| | | | RMV(¬PW) | 0.834 | −0.620 | 0.847 | 0.060 | 0.837 | 0.029 |

win a match a document should become more similar to the winner by adding and not removing terms that were used by the winner (positive ΔMRR of ADD(PW) and negative ΔMRR of RMV(PW)), as well as removing and not adding terms that were not used by the winner (positive ΔMRR of RMV(¬PW) and negative ΔMRR of ADD(¬PW)). These manipulations which do not directly affect the query-document similarity estimates affect other features used by the ranking model (e.g., Entropy).

## 7 CONCLUSIONS

We presented an initial theoretical and empirical study of the strategic behavior of publishers (documents' authors) in query-based ranking competitions. The publishers' goal is promoting their documents in rankings using little available information, mainly about past rankings. Analysis of ranking competitions that we organized revealed that to achieve their goal, publishers were making their documents similar to those ranked the highest in previous rounds. A game theoretic analysis of the competition yielded a result that provides formal support to the merits of this strategy. We also showed that high accuracy prediction of whether a document will be promoted to the first rank in our competitions can be achieved using very few features which quantify document changes.

**Acknowledgments** We thank the reviewers for their comments. This work was supported in part by a Google Faculty Research Award.

## A THE RANKING COMPETITIONS

We next discuss the competition guidelines provided to students (Section A.1), the incentives for participating in the competition (Section A.2), the queries and examples of relevant documents (Section A.3) and the ranking function used (Section A.4).

### A.1 Guidelines

To alleviate the task for students, and to increase their engagement in the competition, the length of *all* documents was limited to 150 terms. Students were instructed to write unstructured plain text documents.

Duplication of other documents (determined based on a bag-of-terms comparison) resulted in the duplicate document being ranked last. The students were permitted to copy parts of other documents from the competition or the Web. Students were guided to write documents of the highest quality avoiding slang and informal language. The use of black hat SEO techniques [16], such as keyword stuffing, was discouraged by telling the students that the ranking function will penalize low quality documents, partly based on human annotations. We informed the students that they could use the provided examples of relevant documents, but that the documents they create need not necessarily be relevant.

### A.2 Incentives

The incentive for participating in the competition was earning extra credit points for the exam. For each query, a student earned two thirds of a point if her document was ranked first for a query in a match. A third of a point was given to all other students competing with respect to the same query (i.e., the same match).

In the first half of the competition many students did not (significantly) update their documents even if these were not ranked first. Therefore, we further incentivized the students by changing the reward mechanism as from the fifth round. The student whose document was ranked first for a query was reworded one point. Students whose documents were ranked second and third were rewarded two thirds and third of a point, respectively. Students whose documents were ranked lower did not receive any credit.

### A.3 Queries and initial relevant documents

We used the titles of 31 topics selected from 1-200 from TREC 2009-2012 as queries. The preference was selecting queries with clear commercial intent, since they were more likely to stir up competition as is the case on the Web. That is, having a document ranked high (first) with respect to these queries should lead to increased (monetary) profits to the document's publisher on the Web. The selected queries focused mostly on topics related to products or services. Examples include "used car parts", "cheap internet" and "gmat prep classes". The queries were randomly assigned to students ensuring that two students will not compete against each other in more than two different matches; the assignments were not changed throughout the competition.

As already noted, for each query we provided a single example of a relevant document. The goal was to provide the students with information regarding the underlying information need as the queries are very short. To produce these relevant documents, we first used the TREC topic description as a query in a commercial search engine. We extracted from the highly ranked documents

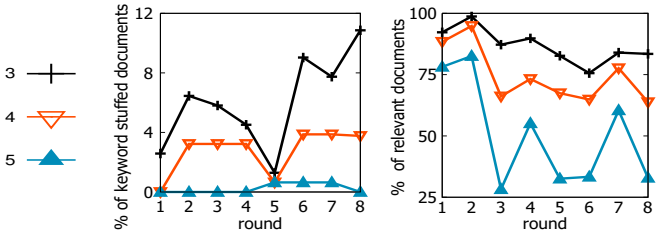


**Figure 3: The percentage of documents annotated as keyword stuffed (left) and relevant (right) by at least 3, 4 or 5 annotators, averaged over queries per each of the eight competition rounds.**

candidate window passages of up to 150 terms. The passages were annotated for relevance by four annotators. We kept extracting passages for each query until a passage was judged relevant by at least three annotators. This passage then served as the initial relevant document example for all students competing for the query.

## A.4 Ranking model

We next describe the ranking model used for all queries in each round of every match in the competition.

*A.4.1 Learning-to-rank.* We used a learning-to-rank (LTR) approach with 25 features to rank the documents. Most of the features (22) were all those used in Microsoft's learning-to-rank datasets[13] for the "whole document" except for the Boolean Model, Vector Space Model and LMIR.ABS features. As noted above, the documents in our competition are unstructured plain text. Thus, all the features are computed only for the entire document. Since documents in our competitive setting are prone to manipulation, we used three additional features which were shown to be highly effective for spam classification [21] and Web retrieval [4]: (i) the ratio between the number of stopwords and non-stopwords in a document, (ii) the percentage of stopwords in a stopword list that appear in the document, and (iii) the entropy of the term distribution in a document [4]. For the two stopword-based features, the list of stopwords was composed of the 100 most frequent alphanumeric terms in the ClueWeb09 Category B corpus [21].

The ClueWeb09 category B dataset with queries 1-200 was used to learn the LTR model. Specifically, the model was applied upon the 1000 documents most highly ranked by using LMIR.DIR, i.e., the negative cross entropy between the unsmoothed and Dirichlet-smoothed (with $\mu = 1000$) unigram language models induced from the query and documents, respectively[14]. We used LambdaMART [29] via the RankLib library[15] to integrate the different features. The number of trees and leaves were selected from $\{100, 250, 500, 750, 1000\}$ and $\{10, 25, 50\}$, respectively. NDCG@5 served for optimization when learning the model. In each round of the competition, we added the (unjudged) documents submitted by students in all matches to the ClueWeb09 Category B corpus to have more updated values of corpus statistics, e.g., inverse document frequency (idf). Yet, we did not re-train the ranker. The Indri toolkit was used for indexing and retrieval[16]. We applied Krovetz stemming upon queries and documents and removed stopwords on the INQUERY list only from queries. The LMIR.JM feature was used with $\lambda = 0.1$; for BM25, we set $k1 = 1.2$ and $b = 0.75$.

*A.4.2 Results diversification.* To encourage students to considerably change their documents rather than introduce minor modifications to the initially provided relevant document, starting from the second round, they were advised to diversify their documents with respect to the relevant document. To further encourage diversification, we applied the MMR method [5] with respect to the initial relevant document $d_{\text{init}}$. Accordingly, the score assigned to document $d$ with respect to query $q$ is $score(q, d) \stackrel{def}{=} \lambda rank(d, LTR) - (1 - \lambda) rank(d, d_{\text{init}})$, where $\lambda = 0.5$, $rank(d, LTR)$ is the rank of $d$ in a ranking of all the documents in a match induced by the LTR method and $rank(d, d_{\text{init}})$ is the rank of $d$ in a ranking created based on the similarity with $d_{\text{init}}$; here, the rank of the lowest ranked document is 1. The similarity with $d_{\text{init}}$ was computed using LMIR.DIR treating $d$ as the query.

*A.4.3 Keyword stuffing.* Keyword stuffing [16], specifically of query terms, is one of the most applicable manipulation approaches the students could employ to promote their unstructured plain text documents in rankings. To avoid rewarding excessive keyword stuffing, and to encourage writing of high quality documents, each document was manually classified as keyword stuffed or not[17]. The annotation was performed via CrowdFlower[18]; each document was judged by five annotators from English speaking countries[19]. The inter-annotator agreement for keyword stuffing, computed using the free-marginal multi-rater kappa measure [24], is 0.88. A document classified as keyword stuffed by at least four annotators was rank-penalized: with probability 0.5 it was swapped with the next document in the ranking. If several consecutively ranked documents were keyword stuffed, then only the lowest ranked document was penalized.

In Figure 3 (left) we present for each round the percentage of documents classified as keyword stuffed by at least three, four or five annotators averaged over queries. We can see a mostly downward trend until the fifth round. In the fifth round we observe the lowest percentage of keyword stuffed documents. Starting from the fifth round the percentage of keyword stuffed documents gradually increases. We hypothesis that in the first half of the competition students' engagement gradually decreased. In the second half, as from the fifth round in which the rewards for having a document ranked high substantially increased, students started using manipulated texts even more so as to have their documents ranked high. In the fifth round, there might have been some confusion due to the introduction of a new reward mechanism.

[13] www.research.microsoft.com/en-us/projects/mslr
[14] We deliberately did not remove suspected spam documents from the initial document ranking, e.g., using Waterloo's spam classifier [7]. This practice allows learning a model using low quality (e.g., spam) documents.
[15] www.lemurproject.org/ranklib.ph
[16] www.lemurproject.org/indri
[17] A document was annotated as keyword stuffed if it contained excessive repetition of words which seemed unnatural or artificially introduced.
[18] www.crowdflower.com
[19] Annotators were also instructed to classify documents as spam if they were hard to understand, non-cohesive, did not make any sense or were useless to anyone seeking information. Yet, none of the documents was classified as spam.

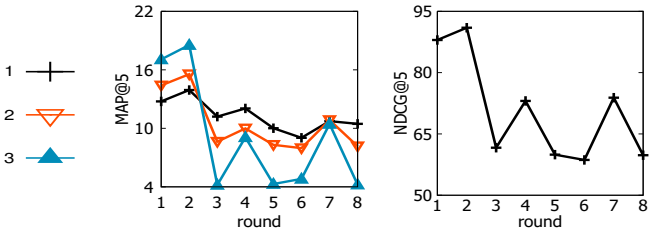


**Figure 4: The MAP@5 (left) and NDCG@5 (right) performance of the ranking induced by the retrieval method in each round. Binary relevance judgments were induced for computing MAP@5 by considering a document relevant if its relevance grade was at least 1, 2 or 3.**

## A.5 Ranking effectiveness

All documents in the collection were judged for relevance. Annotators were presented with both the title and description of each TREC topic, and were asked to classify a document as relevant if it satisfied the information need stated in the description. As was the case with keyword stuffing annotation, each document was judged by five annotators from English speaking countries via CrowdFlower. The inter-annotator agreement rate, computed using the free-marginal multi-rater kappa measure [24], was 0.67. Four-scale graded relevance judgments were generated using the annotations as follows. A document judged relevant by less than three annotators was labeled as non-relevant (0). Documents judged relevant by at least three, four or five annotators were labeled as marginally relevant (1), fairly relevant (2) and highly relevant (3), respectively.

As noted above, to address the potential manipulation of documents by students, the retrieval method used in the competition (i) was based on a learning-to-rank approach with multiple features, (ii) incorporated highly effective document-quality measures and (iii) penalized keyword stuffed documents. Figure 3 (right) presents the percentage of documents classified relevant by at least three, four or five annotators per round averaged over queries. We see that, in general, the percentage of relevant documents decreased over the course of the competition. While many of the documents were judged relevant by at least three annotators, far fewer documents were judged relevant by at least four or five annotators. This finding attests to the negative effects of SEO.

In Figure 4 we present the MAP@5 and NDCG@5 effectiveness of the document ranking induced by the retrieval method in each of the eight competition rounds. We see that the effectiveness of the ranking has gradually decreased over rounds, which can be partially attributed to the fact that fewer relevant documents were generated by students as seen in Figure 3. We also see that in the first two rounds the effectiveness of the ranking was much higher than that in the rounds to follow. We found that in the first two rounds students used the initially provided relevant documents without significantly changing them. After the second round, in which the retrieval method was changed by applying diversification with respect to the given relevant document (see Section A.4.2), students started diversifying their documents by introducing noise, using non-relevant information and applying content manipulation.

## REFERENCES


[1] R. Aumann and M. Maschler. 1995. *Repeated Games with Incomplete Information.* MIT Press.
[2] Ran Ben-Basat and Elad Kravi. 2016. The ranking game. In *Proceedings of the 19th International Workshop on Web and Databases.* 7.
[3] Ran Ben-Basat, Moshe Tennenholtz, and Oren Kurland. 2015. The Probability Ranking Principle is Not Optimal in Adversarial Retrieval Settings. In *Proceedings of ICTIR.* 51–60.
[4] Michael Bendersky, W. Bruce Croft, and Yanlei Diao. 2011. Quality-biased ranking of web documents. In *Proceedings of WSDM.* 95–104.
[5] Jaime G. Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of SIGIR.* 335–336.
[6] Carlos Castillo and Brian D. Davison. 2010. Adversarial Web Search. *Foundations and Trends in Information Retrieval* 4, 5 (2010), 377–486.
[7] Gordon V. Cormack, Mark D. Smucker, and Charles L. A. Clarke. 2011. Efficient and effective spam filtering and re-ranking for large web datasets. *Informaltiom Retrieval Journal* 14, 5 (2011), 441–465.
[8] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. 2004. Adversarial Classification. In *Proceedings of KDD.* 99–108.
[9] Fernando Diaz. 2005. Regularizing Ad Hoc Retrieval Scores. In *Proceedings of CIKM.* 672–679.
[10] Ran El-Yaniv and Mordechai Nisenson. 2010. On the Foundations of Adversarial Single-Class Classification. *CoRR* (2010).
[11] Kfir Eliaz and Ran Spiegler. 2011. A simple model of search engine pricing. *The Economic Journal* 121, 556 (2011), F329–F339.
[12] Kfir Eliaz and Ran Spiegler. 2016. Search design and broad matching. *American Economic Review* 106, 3 (2016), 563–586.
[13] Jonathan L. Elsas and Susan T. Dumais. 2010. Leveraging temporal dynamics of document content in relevance ranking. In *Proceedings of WSDM.* 1–10.
[14] Hui Fang, Tao Tao, and ChengXiang Zhai. 2004. A formal study of information retrieval heuristics. In *Proceedings of SIGIR.* 49–56.
[15] Norbert Fuhr. 2008. A probability ranking principle for interactive information retrieval. *Information Retrieval* 11, 3 (2008), 251–265.
[16] Zoltán Gyöngyi and Hector Garcia-Molina. 2005. Web Spam Taxonomy. In *Proceedings of AIRWeb 2005, First International Workshop on Adversarial Information Retrieval on the Web.* 39–47.
[17] Nathanael Hyafil and Craig Boutilier. 2004. Regret Minimizing Equilibria and Mechanisms for Games with Strict Type Uncertainty. In *Proceedings of UAI.* 268–277.
[18] N. Jardine and C. J. van Rijsbergen. 1971. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval* 7, 5 (1971), 217–240.
[19] John D. Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR.* 111–119.
[20] Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval.* Springer. I–XVII, 1–285 pages.
[21] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. 2006. Detecting spam web pages through content analysis. In *Proceedings of WWW.* 83–92.
[22] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[23] Kira Radinsky and Paul N. Bennett. 2013. Predicting content change on the web. In *Proceedings of WSDM.* 415–424.
[24] Justus J. Randolph. 2016. Online Kappa Calculator (2008). Retrieved February 6, http://justus.randolph.name/kappa. (2016).
[25] 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.
[26] 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.
[27] Marc Sloan and Jun Wang. 2012. Dynamical information retrieval modelling: a portfolio-armed bandit machine approach. In *Proceedings WWW.* 603–604.
[28] Hong Wang, Wei Xing, Kaiser Asif, and Brian D. Ziebart. 2015. Adversarial Prediction Games for Multivariate Losses. In *Proceedings of NIPS.* 2728–2736.
[29] 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.
[30] Grace Hui Yang, Marc Sloan, and Jun Wang. 2016. *Dynamic Information Retrieval Modeling.* Morgan & Claypool Publishers.
[31] Yinan Zhang and Chengxiang Zhai. 2015. Information Retrieval as Card Playing: A Formal Model for Optimizing Interactive Retrieval Interface. In *Proceedings of SIGIR.* 685–694.