The Ranking Game

Ran Ben Basat
sran@cs.technion.ac.il

Elad Kravi
ekravi@cs.technion.ac.il

Department of Computer Science, Technion, Haifa, Israel

ABSTRACT

In the web, page creators often compete for their ranking on relevant search queries, as high ranking attracts users and may lead to increased revenues. The positive effect of this competition is that it encourages page owners to improve their content. However, some techniques used for this improvement, such as *keyword stuffing*, are considered harmful. Current search engine literature largely ignores the interplay between the retrieval system and the pages' content evolvement. This paper studies the above phenomenon.

We model pages as strategic players, competing for a better rank by content manipulations, while the search engine controls the ranking mechanism. We show that such competition may degrade retrieval effectiveness as irrelevant pages tend to outrank better ones. Further, we investigate how the search engine's choice of a ranking scheme may reduce the incentive to manipulate the page content. Finally, we propose a novel ranking solution that empirically minimizes the adverse effect on a real dataset.

1. INTRODUCTION

Users tend to focus most of their attention on the highest ranked results presented by search engines [4]. This is called "trust bias" and it makes the user believe that the higher a page is ranked, the better it answers her information need. Consequently, creators of web pages are motivated to optimize the page content for ranking, as many users may leave the result page while ignoring pages with low rank.In order to achieve higher ranks, web pages adjust their content to be more adequate for specific queries in a process called Search Engine Optimization (*SEO*). Some adjustments (known as *white-hat SEO*) are encouraged by search engines; others, like *stuffing keywords* are discouraged [6].

Stuffing popular search terms may manipulate search engines to retrieve irrelevant pages [14]. For example, consider a web page about issuing a visa to United States. One relevant query is *"issue visa united states"*. By stuffing terms like *"visa"*, *"issue"*, and *"united states"*, the page may obtain higher rank by search engines.

Many retrieval models classify web pages to be either "spam" or "informative". Other models gives score penalties to pages with lower term distribution entropy, which was found correlated with spamming [10]. Most of these models, regardless of the way they treat spam, consider web pages as static [13], i.e., with a content independent of the search engine's ranking. However, in practice sites often put a great effort on SEO and thus their content is highly dependent on the ranking of the search engine. Furthermore, web pages that contain many topics are less likely to be considered as an authority in a specific topic [5]. Therefore, when optimizing their ranking, web pages usually focus on a single topic (or few topics) and adjust their content accordingly.

A major challenge is modeling this complex ecosystem holding the competition between web pages. A model of the system should consider the web pages, and the strategies of their builders for content modifications, in order to achieve higher rank. Choosing how to do so, depends on the SEO cost, and the gain from the improvement in the rank. Another major player in the system is the search engine. Although it cannot directly determine the content of a web page, it can influence it by selecting a ranking procedure that reduces the incentive of pages to SEO themselves. In most retrieval models, the ranking for a given query is determined by a score called Retrieval Status Value (RSV) which is assigned to each document. For example, the *Okapi BM25* similarity [12] is an effective RSV. Consider for example, two web pages, that their builders are competing for higher rank on a specific query. The builder of the lower-ranked page, may decide to SEO its page if it is not too expensive. Should the search engine try to influence this incentive? Can he do it by a choosing different ranking method?

Our work studies the effect of SEO on the relevance of retrieved documents. In order to model the competition of page creators over rank, we consider each web page to be a strategic player with utility derived from its rank. We assume that each page owner chooses a specific query and try to optimize the content of her page for obtaining the highest possible rank.We show that this SEO process, degrades the retrieval relevance. We present a new ranking procedure that is designed to handle the adversarial nature of web documents. Specifically, we introduce randomization to the ranking process as described in detail in section 4. We conducted a set of experiments, measuring the effect of competition over ranks using standard relevance metrics. Our results suggest that relevance is impaired by the competition process. This is important since the effect of such competi-

WebDB'16, June 26-July 01 2016, San Francisco, CA, USA

 ISBN 978-1-4503-4310-7/16/06…$15.00

DOI: http://dx.doi.org/10.1145/2932194.2932201

tion is often considered positive. Further, our initial results suggest that by introducing randomization to the ranking process, search engines may be able to minimize these adverse competition effects.

The rest of this paper is organized as follows: In Section 3 we present our model. Section 4 presents our novel method for ranking web pages. Section 5 presents the implementation of the model. Section 6 presents an evaluation of the model over real world dataset. Section 7 discusses the results and provides an outlook of the study.

2. RELATED WORK

The topic of web spam was studied intensively [13]. Ntoulas et al. [8] downloaded more than 105 million web pages, and pointed out on features found in pages, that can help in detection of spam. They focused on keyword-stuffing, and offered several heuristics, which were used for training a classifier for automatic detection of spam pages. Castillo et al. [3] have classified, with the help of volunteers, $2,725$ hosts with regard to web spam. Each host was classified as "Normal", "Borderline", "Spam" or "Can not classify". They used guidelines given by the main search engines. By their definition, a spam page is one that contains *synthetic text*— "text that does not appear to be natural language, but consists of phrases and words "stitched" together to form meaningless paragraphs...". An example for synthetic text can be a sequence of query terms appended to the original document. Both studies signify the importance of web spam, and support our approach of detecting SEO keyword-stuffing by anomalies in the page content.

Another studied aspect of web spam is the concept of re-ranking documents, boosting pages with lower indication of spam. Raiber et al. [10] presented a retrieval model composed of two stages. In the first stage, documents which has a high surface-level similarity to the query were retrieved. In the second stage, the relevant set was re-ranked, and documents having low inner-document similarity to the relevant set were penalized. Their method was shown to achieve high precision for the task of searching web pages.

All of the studies presented above considered web pages to be static, and did not refer to changes in content of pages over time, an effort constantly done by SEO agents for improving the rank of their page.

Given a document d, a *language model*, M_d, is a generative model describing the probability of a document to generate a given text. The model assigns a probability to each term that appears in the document proportional to its frequency. The *query likelihood (QL)* [9] is the probability that a given query was generated by the document's language model. Documents with high probability to generate the query are assumed to be more relevant to answer its information need. Formally, given a query q and a document d, the query likelihood is defined as $QL(q,d) = \Pr(q|M_d) \triangleq \prod_{t \in q} t|M_d$, where $\{t \in q\}$ are the query terms. Some documents may be relevant for the query even if they are missing some of the query terms. This is usually resolved by *smoothing*, which assigns a non-zero probability to terms that do not appear in document. For example, Laplace smoothing [7] concatenates to the each document one occurrence of each lexicon term. For example, the probability of a term t given a language model M_d, the probability of the terms calculated using Laplace Smoothing will be $\Pr(t|M_d) = \frac{1+\#t}{|d|+|\Sigma|}$.

In our model, we simulate the contest between web pages as a game, where each page is a rational player trying to maximize its utility. A competition between web pages was introduced by Ben-Basat et al. [1]. They examined the search engine ability to influence the *topics* of documents, generated by websites in order to satisfy the information need of the society. They considered each document as a strategic player trying to achieve the maximum rank. The work showed that the traditional Probability Ranking Principle [11], which maximizes the retrieval effectiveness by ranking documents in decreasing order of relevance probability, is not optimal when documents are not static. Instead, it presents a *probabilistic* ranking, which randomizes the ranks of documents with similar relevance score. Their results show that such ranking may bias the documents' incentives towards a more varied topic collection that increases the social-welfare of the users. Our work takes a different approach as we introduce a *cost* for SEO that limits the content modifications. Also, in our model pages differ from each other in their initial content.

3. MODEL

We now present our model describing a competition between web-pages and its effect on the their content. We denote the set of all possible documents by $\mathcal{D}$ and all queries by $\mathcal{Q}$. The process starts with a given query $q \in \mathcal{Q}$ and n competing documents $d_1, \ldots, d_n \in \mathcal{D}$. Both the documents and query are given in a bag-of-words representation over a lexicon Σ (i.e., $\mathcal{D} = \mathcal{Q} = \mathbb{N}^{\Sigma^*}$) with no proximity information, meta data or any additional information. We consider retrieval methods that are based on a Retrieval Status Value (RSV) function. That is, we assume the existence of a score function $RSV : \mathcal{D} \times \mathcal{Q} \to [0,1]$ which estimates the relevance of this document-query pair. The RSV function is known both to the search engine and the documents, allowing documents to calculate the competitors' RSV score. This assumption can be relaxed to the more realistic scenario where pages are unaware of the RSV function, but can observe the current ranking by querying the search engine. Alternatively, owners can calculate an approximated RSV value by using publicly known functions. Traditionally, documents are ranked by a decreasing order of their RSV scores; in Section 4 we propose an alternative mechanism.

3.1 Competition Between Documents

We consider the competition between n documents, where each document is considered to be a strategic player that wishes to maximize its profit by being ranked as high as possible. The profit of a player is determined by a function $P : \{1, \ldots, n\} \to \mathbb{R}^+$ and is identical for all players. This means that the profit of the player ranked at the i'th position is $P(i)$. We assume that P is monotonically non-increasing. An example for such function is *reciprocal ranking*, in which the profit gained from the i^{th} rank is $1/i$. For example, the profit from being ranked first is 1.

The set of optional actions for each player contains all possible documents, i.e., each document can be modified arbitrarily. Nevertheless, we associate a cost for the SEO process, imitating an agent that charges per amount of work required to modify the document's content for achieving higher rank. This is modeled by a function $C : \mathcal{D} \times \mathcal{D} \to \mathbb{R}^+$. That is, for modifying d_i to d'_i, player i pays $C(d_i, d'_i)$. Denoting the rank of d'_i as $r(d'_i)$, the *utility* of the SEO action is defined

to be $U_i(d'_i) = P(r(d'_i)) - C(d'_i, d_i)$, meaning, the profit minus the cost of SEO. We assume that $\forall d \in \mathcal{D} : C(d, d) = 0$, i.e., there is no cost for not SEOing the document. Notice that the extent to which the content is modified depends on the difference between the profit and cost. For example, if the cost for modification is high almost no action will take place. In contrast, high profits will lead players to perform excessive content manipulations. As SEO is prominent in the web, we assume that the competition is profitable.

In our work, we focus on keyword stuffing. That is, we assume that pages do not remove parts of their content but rather try to optimize the rank by stuffing additional words. This means that the cost function C only applies to pairs (d, d') such that $d \subseteq d'$. Note that this can be modeled by associating a cost higher than $P(1)$ for all other changes, making them unprofitable.

3.2 Best Response

While there are many possible outcomes to this competition, we consider a *best-response dynamics* played by the documents. This means that each player, according to some pre-defined order, observes the set of all documents and maximizes its utility by modifying its content. Note this is a large set of documents (exponential in the number of stuffed terms), that practically will have to be narrowed.

The process of content modification continues iteratively until convergence, where no document can profit from further SEO. Such iterative process closely represents reality; while the RSV function is usually not known, pages consider the search engine ranking at each point and can observe whether a given modification improved their ranking. Finally, we assume that the profit is payed after each iteration, which means that pages need not consider future rankings for choosing their action.

3.3 Competition Example

Consider the lexicon $\Sigma = \{a, b, c, d\}$, the query $q = \{a, b\}$, the initial documents $d_1 = \{a\}$ and $d_2 = \{b, c\}$. Furthermore, assume that the RSV function used is *query likelihood* [9] after applying Laplace smoothing. The profit is defined such that pages benefit 1 from being ranked first (i.e., $P(1) = 1, P(2) = 0$); and that the SEO cost of *adding each word* is 0.75. Other modifications are not considered (i.e., have cost higher than 1). The initial *RSV* value for each of the documents is:

$$RSV(d_1, q) = \Pr[a|M_{d_1}] \cdot \Pr[b|M_{d_1}] = (2/5) * (1/5) = 2/25$$
$$RSV(d_2, q) = \Pr[a|M_{d_2}] \cdot \Pr[b|M_{d_2}] = (1/6) * (2/6) = 2/36$$

M_d denotes the *language model* of a document, in our example d_1 contains the single term 'a', hence $\Pr(a|M_{d_1}) = \frac{1+\#a}{|d_1|+|\Sigma|} = \frac{2}{5}$, $\Pr(b|M_{d_1}) = \frac{1+\#b}{|d_1|+|\Sigma|} = \frac{1}{5}$, etc. Being second, d_2 would like to improve its rank. The best action for d_2 is stuffing the term a, which raises its *RSV* score to $(2/7)*(2/7) = 4/49 > 2/25$. Choosing to optimize, d_2 incurs a cost of 0.75 which is worthwhile as the change in profit is $P(1) - P(0) = 1 > 0.75$. Next, d_1 observes that it is no longer ranked first and modifies its content to $\{a, b\}$, giving it an *RSV* value of $(2/6) * (2/6) = 4/36 > 4/49$. Finally, d_2 realizes further promotion is not worth the cost associated with the SEO process, as reaching the first rank now requires stuffing multiple words, which costs more than 1, the profit difference. Thus, no agent has an incentive to take further action and the competition ends. The final documents are then $d_1 = \{a, b\}$ and $d_2 = \{a, b, c\}$.

Algorithm 1 Probabilistic Ranking

```
1: function RANK_ρ(S)
2:     U ← S
3:     r ← 1
4:     while U ≠ ∅ do
5:         m ← max_{d∈U}{RSV(d)}
6:         L ← {d ∈ U | RSV(d) ≥ m · ρ}
7:         ℓ ← RandomElement(L)          ▷ Uniform sample
8:         R[ℓ] ← r
9:         r ← r + 1
10:        U ← U \ {ℓ}
11:    end while
12: end function
```

4. PROBABILISTIC RANKING

Traditionally, documents ranking is done deterministically by the order of their RSV scores with respect to the query. Here, we expand the probabilistic model proposed in [1] to rank more than two documents. The goal of introducing randomization to the ranking process is decreasing the incentive of documents to stuff keywords as the profit from this is not guaranteed.

4.1 Ranking Process

We now describe our stochastic ranking process, *probabilistic ranking*. We denote the set of all competing documents by S. Intuitively, we rank documents one at a time starting with the top position. At any rank, we consider as candidates all documents whose RSV score is at least a ρ fraction of the highest RSV of any unranked document. Out of this documents set, we pick one document uniformly at random. Doing so, documents may have lower incentive for stuffing, since improving the RSV score beyond that of another document does not promise increased utility. By limiting the minimal value of ρ, we ensure that inferior documents will not outrank significantly better ones. Similarly, high ranked page is guaranteed to be ranked higher than any page whose quality is significantly lower.

$\rho \in [0, 1]$ is a parameter of the algorithm that controls the randomness level of the model that we later on examine in our evaluation. Out of the candidates set, a single item is chosen uniformly at random and is placed at the highest available rank. The process continuous until all documents are ranked. The output is a ranking $R : S \to \{1, 2, \ldots, |S|\}$. A pseudo-code of the process also appears in Algorithm 1. We start with all documents unranked; at first, the algorithm finds the highest RSV score, m, among the non-ranked documents. Next, it computes the set L which contains all documents whose RSV score is at least $\rho \cdot m$. Then, a single candidate is chosen uniformly out of the L. The process continues until all the documents are ranked.

4.2 Ranking Example

Consider a probabilistic ranking function with $\rho = 0.75$. Also, assume that pages with RSV scores of $1, 0.8, 0.6$ were retrieved for a query. The highest RSV score of any unranked page is $m = 1$, and thus the probabilistic ranking uniformly selects one of $1, 0.8$; the third document is not considered as its RSV score is smaller than $m \cdot \rho = 0.75$. The set of pages that compete over the second rank depends on

the top ranked document's identity. If the page with RSV 1 was ranked first, m becomes 0.8 and thus the second position may be 0.8 or 0.6. In contrast, if 0.8 was ranked first, the only candidate for the second rank is 1, as $\frac{0.6}{1} < 0.75$.

5. SIMULATION SYSTEM

We now describe the implementation of our model, that is later used for the experimental evaluation. The system is composed of several modules interacting with each other as shown in Fig. 1.

5.1 Competition, Round and Iteration

The competition module gets as an input a query and a set of documents. Its output is a set of modified documents, which are the outcome of the competition. The competition progresses in rounds; at every round, each document acts once by a predefined order. Specifically, we set this order such that the document that had the lowest RSV score at the end of the previous round plays first. Then, the second lowest RSV document acts and so the process continues. After the highest RSV score document of the previous round played, the round ends and the order for the next round is computed. We assume that at each turn, the acting page plays its *best response* and the overall competition is then a best response dynamics process.

While in real life a document may compete over several queries, our model we relax this to a single query. Also, the playing order may not keep on fairness, as different players may have different SEO budgets. Finally, there might be a cost for evaluating SEO actions, in addition to the cost of the chosen one. In future work, we plan to address these issues and study their effect on the competition results.

5.2 Best Response Dynamics

For computing the playing document's action, the system searches for the stuffed document that maximizes the utility. Since the number of possible documents is large, we limit this search in several ways. Given a query, each player considers stuffing only query terms, which are usually most effective in raising the RSV score. Albeit, this still does not allow efficient best response computation; therefore, we limit the number of stuffed terms per iteration by a parameter k. Denoting n the number of query terms, there are $CC_n^k = \binom{n+k+1}{n}$ potential moves. We limit the number of terms by setting $k = 3$, allowing feasible calculation of the best response.

In order to further reduce the number of options, we apply a greedy approach. Iteratively, we greedily add the term which locally maximizes the RSV score. The process ends once we added k terms and returns the revised document which maximizes the utility, or the original document if no improvement was found. Note that calculating the utility for the probabilistic ranker requires additional computation as described in Section 5.3 below. The pseudo code for computing the SEO (under the greedy approach) taken by the document appears in Algorithm 2. The algorithm gets as input the initial document d_i, the query q. The while loop in lines 5 to 12 computes the best modification to d_i. In line 6 the algorithm perform the local search presented above.

5.3 Utility Calculation

The ranking module gets as an input a set of documents with their RSV scores, and ranks them. Using these ranks

Algorithm 2 GreedyBestResponse

```
1:  function SEO_k(d_i, q)
2:      bUtil ← 0                          ▷ The best utility so far
3:      bCand ← d_i                        ▷ The best candidate so far
4:      d ← d_i                            ▷ Our current candidate
5:      while |d| − |d_i| < k do
6:          t = arg max_{t∈q} RSV(d ∪ {t})
7:          d ← d ∪ {t}
8:          if U(d) > bUtil then           ▷ Better candidate
9:              bUtil ← U(d)
10:             bCand ← d
11:         end if
12:     end while
13:     return bCand
14: end function
```

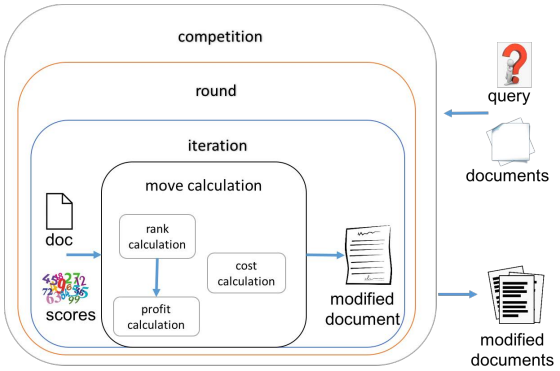

Figure 1: A schema of our system, denoting the competition over a single query.

we calculate the profit and the resulting utility. Unlike the deterministic ranking which is calculated in a single iteration, the probabilistic ranking is evaluated throughout a set of simulations. Notice that the rank of a page is now a random variable; hence, players aim to maximize the *expected utility*. In each simulation we apply Algorithm 1 for calculating the ranking, and then calculate the profit and the utility. The final utility is then the average over all simulations.

We used the open source library of Apache Lucene [2] for storing and manipulating the corpus, calculating RSV scores, and tracking the content of the documents throughout the competition. Our system is implemented in Java.

6. EXPERIMENTAL EVALUATION

We now describe the experiments conducted for testing the model and the proposed probabilistic ranking.

6.1 Dataset and General Settings

In the experiments, we used the AP News & Wall Street Journal dataset [15] which includes news articles from the AP news system and TREC's relevance judgments. The judgments are composed of information needs represented by *queries* and the set of articles satisfying each need.

We set the *profit* function to be the *reciprocal ranking*, and the SEO *cost* as a fixed cost of 0.05 per stuffed term; i.e., the value for the $i'th$ ranked document is $1/i$, and the SEO cost of stuffing j terms is $0.05 \cdot j$. For each query, we considered a competition of 20 pages that was terminated

after 10 rounds if it did not converge by then. We used the BM25 [12] as a deterministic ranking function benchmark. For evaluating our proposed probabilistic ranking, we used a Monte-Carlo simulation of 10 thousand iterations.

6.2 Relevance Experiments

In order to measure the relevance, we used the Precision at k metric for $k \in \{3, 5, 10\}$ denoted as $P@3, 5, 10$ in Table 1. Precision at k is the percentage of relevant documents among the top k positions. Additionally, we measured the Mean Average Precision (MAP). Average Precision (AP) is the average $P@k$ for all positions k in which relevant documents are ranked. The MAP metric calculates the average AP over all the queries. For each query, we retrieved an *initial set* containing the 20 documents ranked first by the BM25 similarity [12]. TREC human judges supplied the set of relevant documents that satisfy the *information need* expressed by each query. Only queries with at least one relevant document among the initial set were considered. This left us with 79 queries.

We simulated the competition using both deterministic and probabilistic ranking and measured the relevance of the ranking over the modified documents. We also computed the relevance metrics for the initial documents set in order to understand the effect of competition.

Our assumption is that the relevance, or lack thereof, of a document to the query need is not changed by the content modifications made due to the competition. For relevant documents, adding keywords to the content does not impair the relevance as it surely still satisfies the information need. On the other hand, irrelevant documents are assumed to stay so, since if the writer of the content could not satisfy the information need at first, stuffing query terms repeatedly will not add a coherent answer that can change this.

The results, presented in Table 1, show a decrease in relevance across all metrics. The first row shows the stats achieved by BM25 for a static corpus. The second is computed for the modified documents under a deterministic ranker competition, while the last employs probabilistic ranking using $\rho = 0.9$. The decrease in relevance can be explained as irrelevant pages ranks improve throughout the competition. Also, the probabilistic ranking is comparable to deterministic. Unfortunately, due to the small size of the corpus, these results are not statistically significant.

	P@3	P@5	P@10	MAP[1]
Initial Set	**0.5105**	**0.4835**	**0.4734**	**0.0655**
Det. Competition	0.4557	0.4481	0.4341	0.0591
Prob. Competition	0.4492	0.4490	0.4449	0.0590

Table 1: Comparison of the relevance metrics before and after the competition.

6.3 Keyword Stuffing Experiment

As mentioned above, keyword stuffing is considered a harmful method of SEO and may result in inferior retrieval results in addition to a negative impact on the text coherence. In this experiment, we measured the affect our randomized ranking process has on the amount of keyword stuffed.

[1]The MAP values are unusually low for all cases, as we only retrieve 20 documents rather than 1000.

The competition contained 20 pages which tried to optimize their rank. The initial state of every document was the actual text appeared in the AP News dataset [15]. We limited the competition to 10 rounds to ensure termination; that is, every player had 10 chances to optimize its ranking.

The results as presented in Figure 2, show an interesting behavior of the probabilistic ranking. When the threshold ratio ρ is large, the total number of stuffed terms *increases*, compared to the deterministic ranking. For lower values of ρ, we observe a decrease in the number of stuffed words; for values of ρ smaller than 0.95 we get less stuffing than the amount created when pages were ranked deterministically.

The decrease in number of stuffed terms as ρ decreases can be explained since stuffing is less profitable as more pages randomly compete for high ranks. A possible explanation for the increase in stuffing for large ρ values is the incentive of highly-ranked pages to stuff terms in order to avoid the competition, which does not occur in deterministic ranking. This behavior can be furthered illustrated by the following example: Consider two pages with similar, but non-identical RSV values. When deterministic ranking is used, the page with the higher RSV value is ranked first and will not gain from stuffing additional terms. However, if the RSV ratio is smaller than ρ, a coin flip will determine the identity of the higher-ranked page. To prevent the risk of being ranked second, higher-ranked page may choose stuff keyword for boosting its score outside of the ρ-threshold range. In contrast, when the ratio ρ is low, optimizing the content of the higher-ranked page for avoiding the risk of being ranked second may prove too costly, and no SEO will take place.

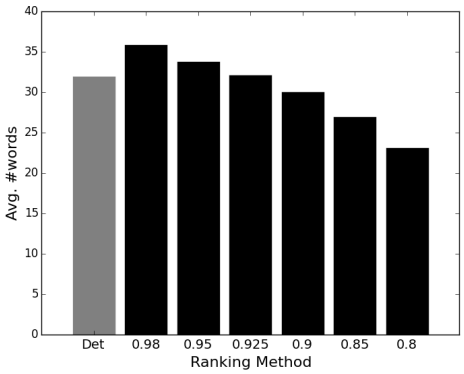

Figure 2: Comparison of the total number of stuffed terms for deterministic ranking and for various values of ρ in probabilistic ranking.

We conclude that the threshold ratio should be carefully chosen, and potentially tailored to the specific RSV function and/or dataset. If we set the threshold too high, we may actually increase the competition and incentivize higher levels of keyword stuffing. On the other hand, setting ρ too low may degrade the retrieval effectiveness as it may rank pages which are less likely to be relevant first due to randomness.

6.4 Convergence Experiment

After establishing the negative effects of competition between pages, we wish to measure the ranking mechanism's influence on the convergence of such competition. Intuitively, excessive competition between pages has negative

side-effects — the retrieval effectiveness might be reduced and the pages get less coherent. In addition, we wish to reduce the effort put into SEO that is not focused on improving the quality of a page.

In this experiment, we examine the possible effects the choice of a ranking method has on the convergence of page competition. We say that a competition converges if no page has an incentive to further modify its content. That is, no page could attain larger payoff by performing the changes required for reaching higher rank.

To ensure termination, we limit the competition to 10 rounds as before, but this time we observe the actual number of round till termination.

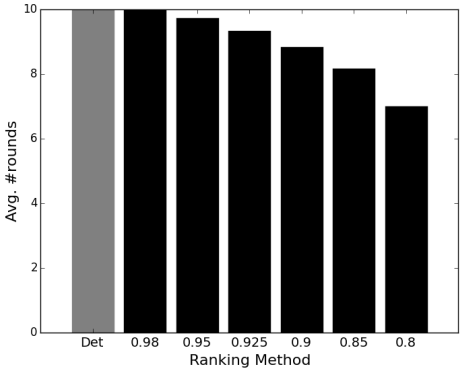

Figure 3: Comparison of the average number of rounds it take for a competition to converge for deterministic ranking and for various values of ρ in probabilistic ranking.

The results, shown in Figure 3, suggest that by introducing randomness into the ranking process, the search engine could limit the extent in which a competition takes place. As in the previous experiment, this shows the virtues of probabilistic ranking and suggests that slight randomization could improve various aspects of the retrieval.

7. DISCUSSION AND CONCLUSIONS

In this section we conclude our work, and discuss potential implications, our limitations and the future work.

7.1 Implications

In this work, we have studied the effect of web pages stuffing query terms while striving to optimize their ranking. Our results suggest that when allowed to stuff keywords, pages modify their content to achieve better ranking; by that, the retrieval relevance may be degraded. We proposed a randomized ranking function, which randomly selects the next page to rank from the set of 'good' documents that have not been ranked yet. Our experiments show that such function could help reduce the incentive of web pages to spam, and by that reduce side effects of the competition. Using our approach, search engines may succeed in reducing web spam and potentially improve relevance.

Our research gives rise to the study of dynamic changes made by SEO in the web. Our system allows researchers to evaluate different manipulation schemes and ranking mechanisms simply by implementing the relevant API.

7.2 Limitations

A limitation of this work is the relatively small number of information needs with relevance judgments included in the AP dataset. Also, our work offers a greedy based approach for calculating the best response of documents. However, our greedy approach does not guarantee that the document performs the best response.

7.3 Future Work

While we attempt to simulate a competition scenario which is frequent in the web, we deliberately chose a dataset that does not contain web pages. Since our documents are news articles, we treat them as non-optimized, and use them to simulate the competition on a clean dataset. Using web pages to evaluate our work we would be facing a potentially already-optimized content, thus making further competition irrelevant. We plan to examine our system over a dataset of web pages to complete the picture.

Another direction for future research could be implementing other spam-aware ranking functions, such as the one introduced in [10]. Finally, our framework could be extended to other SEO operations, such as link farming (which requires extending the bag of words representation of pages).

Acknowledgments

We thank Roy Friedman, Yaron Kanza, Benny Kimelfeld and Oren Kurland for their helpful comments. The work was partially funded by the Technion-HPI research school.

8. REFERENCES

[1] R. Ben Basat, M. Tennenholtz, and O. Kurland. The probability ranking principle is not optimal in adversarial retrieval settings. In *ICTIR*, pages 51–60. ACM, 2015.

[2] A. Białecki, R. Muir, and G. Ingersoll. Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval*, page 17, 2012.

[3] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna. A reference collection for web spam. In *SIGIR*, volume 40, pages 11–24. ACM, 2006.

[4] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, pages 154–161. Acm, 2005.

[5] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604–632, 1999.

[6] R. A. Malaga. Search engine optimization—black and white hat approaches. *Advances in Computers*, 78:1–39, 2010.

[7] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[8] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *WWW*, pages 83–92. ACM, 2006.

[9] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281. ACM, 1998.

[10] F. Raiber, O. Kurland, and M. Tennenholtz. Content-based relevance estimation on the web using inter-document similarities. In *CIKM*, pages 1769–1773. ACM, 2012.

[11] S. E. Robertson. The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304, 1977.

[12] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, pages 109–109, 1995.

[13] N. Spirin and J. Han. Survey on web spam detection: principles and algorithms. *SIGKDD Explorations Newsletter*, 13(2):50–64, 2012.

[14] W. Wang, G. Zeng, and D. Tang. Using evidence based content trust model for spam detection. *Expert Systems with Applications*, 37(8):5599 – 5606, 2010.

[15] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *SIGIR*, pages 81–88. ACM, 2002.