A Two Player Game To Combat Web Spam

Michelle Goodstein*
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213-3891
mgoodste@cs.cmu.edu

Virginia Vassilevska
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213-3891
virgi@cs.cmu.edu

ABSTRACT

We present a novel approach to combating web spam. In the spirit of Luis von Ahn's games with a purpose, we propose using a two player game to identify spam pages within search results. Our game asks users to classify a page as either highly relevant to a query or not relevant to a query, with the option of passing. We use data from the game as the input to a simple voting algorithm which determines whether a page is spam. We show that the best strategy for users playing the game for fun is to answer truthfully, and that spammers have difficulty obstructing the game. Our system can also be generalized and used to obtain relevancy feedback in information retrieval settings.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms

Algorithms, Games

Keywords

Web Spam, Content Analysis, Games With A Purpose

1. INTRODUCTION

Placement in search result pages is a lucrative business. Search engines try to provide the most relevant pages for queries at the top of the results page. Webmasters, knowing that high placement translates into profit, want their pages to be ranked as highly as possible. **Web spam** occurs when webmasters manipulate their sites to take advantage of search engines' ranking algorithms. Their goal is to get their sites ranked higher than the sites' relevance to a query merits. Web spam is abundant, even among the top query results: an example for query "nokia motorolla" follows at the end of the section.[1]

Current methods for combating web spam treat it like an arms race: researchers strive to create new algorithms to detect web spam, while **spammers** work on ways to get by these techniques [1, 2]. Little has been done to provide theoretical performance guarantees for existing methods. In this paper we present a scheme which uses a two player game to collect virtual votes for whether a page is web-spam. These votes are then processed in a way which gives provable performance guarantees. We show that the game is strategy-proof for players interested in scoring well and that we can detect and prevent all others from interfering with the scheme. Our system has further applications in the field of information retrieval where, with minor modifications, it can be used to help provide feedback on a document's relevancy to a query.

In Section 2 we provide an overview of the problem. Section 3 examines related work. Our web spam detection scheme is detailed in Section 4. The proof that our scheme is strategy-proof to rational users is presented in Section 5. We demonstrate resilience to adversaries in Section 6. Finally, we show a possible generalization to information retrieval and relevancy judgments in Section 7.

> FREE RINGTONES - FREE RINGTONES
> ... FREE SANTO RINGTONES LOGOS NOKIA MOTOROLA FREE RINGTONES RING TONES FREE ... POLYPHONIC RINGTONES ON MY MOTOROLA T FREE SEAN PAUL RINGTONES FOR NOKIA ...

2. OVERVIEW OF THE PROBLEM

Most search engines have good algorithms for ranking pages. However, these algorithms occasionally make mistakes, such as ranking a page higher than the average user would like. Much research is devoted to improving page ranking and spam detection, with considerable focus on **link spam**. Link spam occurs when a target page's rank is increased via many

*This research was sponsored in part by the Henry Luce Foundation through a Clare Booth Luce Graduate Fellowship, in part by National Science Foundation (NSF) grant no. CCR-0122581 and in part by the Army Research Office through grant number DAAD19-02-1-0389 ("Perpetually Available and Secure Information Systems") to Carnegie Mellon University's CyLab. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Henry Luce Foundation, the NSF or the US government.

[1]Sixth Google result on 2/6/07, from `www.directory.lmc.edu/public_facilities_viewindividual-52.php`

other pages pointing to each other and to the target. Our goal is, given an already prepared ranking of webpages, to identify pages that are likely spam and "fix" the search results. Our approach involves collecting votes from a sample of individuals. These votes should tell us whether a page is spam or not with respect to a query. When a sufficient number of people have responded and enough have voted the page spam, we remove it from the ranking. One objection that might be raised is the inability of our scheme to detect so-called **honey-pots**–spam pages with content that appears legitimate to the user–or that we will mislabel a legitimate page as spam since we focus on content. Note, however, that our scheme is designed to act in concert with other algorithms. Until we have algorithms that can prevent all spam, some spam pages will persist. Our approach helps ensure that if any page evades link-based detection it must be relevant to the user. We begin with definitions.

DEFINITION 2.1. *Let $\mathcal{Q}$ be a fixed query, and let N be the number of search results for $\mathcal{Q}$. Define $R_{\mathcal{Q}}$ as the $N \times 1$ vector representing a ranking of result pages for $\mathcal{Q}$.*

DEFINITION 2.2. *Let $C_{\mathcal{Q}}$ be an $N \times 2$ matrix such that $C_{\mathcal{Q}}[\mathcal{P}, 0]$ is the number of votes for page $\mathcal{P}$ as highly relevant with respect to query $\mathcal{Q}$, and $C_{\mathcal{Q}}[\mathcal{P}, 1]$ is the number of votes for $\mathcal{P}$ as irrelevant with respect to $\mathcal{Q}$.*

An agent is **honest** if he truthfully classifies page-query pairs, or passes when unsure. By definition, an honest agent cannot have a stake in any page-query pair. Define an **overwhelming majority** as agreement among at least 90% of users. A page is **ranked highly** if it is among the top 10% of the results for a query.

DEFINITION 2.3. *A web page $\mathcal{P}$ is considered to be **web spam** if it is ranked highly in $R_{\mathcal{Q}}$ but an overwhelming majority of honest agents believe it is not relevant to $\mathcal{Q}$.*

As this is still an emerging field of study, there are several definitions currently in use for web spam. Most of them [2, 3, 4, 5, 6, 7, 8, 9] define web spam in terms of actions by a webmaster. Our definition differs by immediately offering measurable criteria, as we will show we can simulate honest users. Our definition also has an advantage as it aligns closely with the concept of **relevancy judgments** or **relevancy feedback** in information retrieval [10].

Our goal is to create a new ranking $R'_{\mathcal{Q}}$ by removing some pages from $R_{\mathcal{Q}}$ using information from $C_{\mathcal{Q}}$.

DEFINITION 2.4. *Let $\hat{R}_{\mathcal{Q}}$ be the ranking created by deleting from $R_{\mathcal{Q}}$ those pages that are voted irrelevant by an overwhelming majority in $C_{\mathcal{Q}}$, where only feedback from honest agents is used in the formation of $C_{\mathcal{Q}}$.*

DEFINITION 2.5. *Let $R'_{\mathcal{Q}}$ be the ranking created by deleting from $R_{\mathcal{Q}}$ those pages that are voted irrelevant by an overwhelming majority in $C_{\mathcal{Q}}$.*

In our scheme, non-adversarial voters are expected to vote honestly, meaning R' and $\hat{R}$ should be similar.

3. RELATED WORK

Gyongyi, Garcia-Molina and Pedersen [4] present the link-based TrustRank algorithm. Similar in methodology to PageRank [11], TrustRank works by propagating trust through the web graph. Gyongyi *et al.* [6] propose another link-based algorithm which estimates spam mass, a metric of how much the PageRank of a page is affected by links from spam pages. The only human component in both algorithms is the seed set evaluation. Wu, Goel and Davidson [7] present a modified version of TrustRank, designed to patch some vulnerabilities within TrustRank. In a similar vein, Krishnan and Raj [12] propose a link-based system that propagates distrust.

Da Costa Carvalho et al. [13] focus on the link graph model of the web, but at the site level instead of at the page level. Their algorithm attempts to detect suspicious links so these can be ignored when PageRank [11] is run. Fetterly, Manasse and Najork [2] examine statistical ways of analyzing URLs, host names, the web graph and individual page content. Ntoulas *et al.* [14] develop other content-based methods of deciding whether a page is spam. *Cloaking spam*, where one page is shown to a search engine and another to a user for the same, as well as *redirection spam*, where the page shown to the search engine performs an automatic redirect when visited by a user, are also being targeted with specific detection algorithms [15, 8]. Recently, Caverlee and Liu [16] and Caverlee *et al.* [17] have proposed techniques that incorporate more theoretical analysis. All these methods focus on automation.

Unlike previous work, we constantly solicit user feedback. By collecting user data, we can use statistical methods to derive whether the average user thinks a page is spam. If so, our algorithm will succeed as we define spam in terms of public opinion. Our criteria for web spam are different and perhaps more complex from those given to automated algorithms– there may be factors visible to a human that a computer cannot easily see that allows a human to classify a page as spam where a program cannot. As a result, it is hard for an adversary to fool our scheme. The only adaptation we offer an adversary is to make a page more relevant to a query. In addition, our performance guarantees and the assumptions they rest upon are explicit; we can and do quantify them in the course of this work.

4. A WEB SPAM DETECTION SCHEME

Motivated by the work of Von Ahn and Dabbish [18] in using two player games to solve hard AI problems, we investigate whether similar work is possible for web spam. Our aim is to create a (hopefully) fun two player game which works in addition to automated web spam detection techniques. The game's purpose is to collect votes from the population on whether a webpage is spam with respect to a query. Our final goal is to use a simple voting algorithm to decide whether to move a page down in the rankings. There are easier ways to collect votes. For example, we could let users classify search results as relevant or irrelevant while searching. Since the user picks and classifies the page-query pair, spammers can vote their pages as highly relevant, and other adversaries can

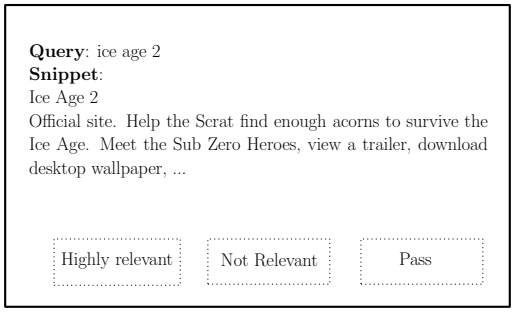

Figure 1: A potential highly relevant (test) question, with query ice age 2 and snippet text from the first Google result for "ice age 2" on 11/10/06

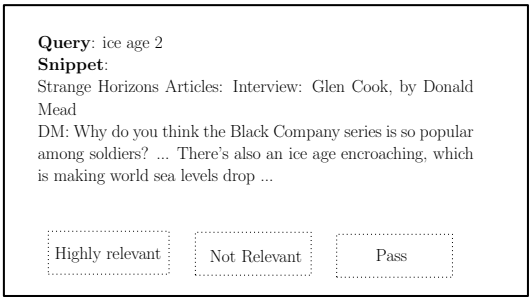

Figure 2: A potential not relevant test question, with presented query "ice age 2" and real query "company cook 2 over age ice". Snippet text from the first Google result for "company cook 2 over age ice" on 11/10/06

vote legitimate pages as irrelevant by repeating their search and voting. By using a game to generate the votes, we avoid these issues.

4.1 A web spam game

Our scheme is based on several crucial assumptions. First, we assume sufficient data storage availability. This seems reasonable since Google Personalized Search allowed users to vote on page-query pairs [19]. We assume that the page ranking $R_{\mathcal{Q}}$ obtained from an external source is approximately optimal. The ranking $R'_{\mathcal{Q}}$ our game should output is supposed to be a "better" version of $R_{\mathcal{Q}}$. However, while there may be errors scattered throughout the original ranking, we only care about results that the user will actually see (say, the first 10 or 20) per query. Therefore, $R'_{\mathcal{Q}}$ only needs to be more accurate for highly ranked items. The goal is to construct the new ranking $R'_{\mathcal{Q}}$ by removing highly ranked items from $R_{\mathcal{Q}}$ that most people rate spam.

4.2 Game description

To prevent dishonest users from choosing the page-query pair they want to affect, we control $\mathcal{Q}$ and $\mathcal{P}$ ourselves. We design our game as a series of s independent questions to pairs of users. Each pair is allotted at most t time units for each question, and T time units for the entire game. Pairings are assigned at random at the beginning of the game, and change with each iteration. Each question consists of a query $\mathcal{Q}$, a short snippet s_p from a randomly drawn page $\mathcal{P}$ in $R_{\mathcal{Q}}$, and three options: "Highly relevant", "Not relevant", and "Pass". For a visualization of the game, see Figure 1. Players get one attempt to answer the question; once they choose an option, they cannot change their minds. Users cannot see what rank $\mathcal{P}$ has in $R_{\mathcal{Q}}$ currently, or the web page URL. The snippet s_p is a small, representative piece of text from page $\mathcal{P}$ [Figure 1]. A key assumption for the game is that s_p represents the page well and that the algorithm for its creation is hidden from the users. Hence we require that a snippet employed by a search engine cannot be used for a page-query pair, as this exposes elements of the algorithm to a potential spammer.

If a user's answer on a question matches that of his partner and neither passed, we increase the user's score by m points. We call this event a *match*. A *mismatch* occurs when neither player passes and their answers disagree. We decrease a player's individual score n points for a mismatch. Zero points are allotted for a pass. Without loss of generality, set $m = 1$ and $n = 1 + \varepsilon$. We later show that $\varepsilon > 0$ is a necessary condition for a game where the dominant strategy is to play honestly. In each round of the game, we incorporate K **test questions** (divided among relevant and non-relevant page-query pairs) for which we know the correct answer. These test questions are designed to be indistinguishable from other questions. A user who answers some test questions correctly and none incorrectly is awarded a bonus $\mathcal{B}$ according to the number of test questions answered correctly, but is not told which questions were test. Table 1 shows an example of how to score a game.

	Q1	Q2	Q3	Q4	Q5	Bonus
Test(X/R/N)	X	R	X	X	NR	—
Player 1	R	NR	NR	P	R	0
Player 2	R	R	NR	NR	P	$\mathcal{B}(1)$
Match Points	$+1$	$-(1+\varepsilon)$	$+1$	0	0	—

Table 1: A scoring example. P=pass, R=highly relevant, NR=not relevant, X=not a test question. Score(Player 1) $=1-\varepsilon$, Score(Player 2) $= 1-\varepsilon+\mathcal{B}(1)$.

4.3 Generating test questions

Our results depend on having many test questions distributed roughly equally among highly relevant and not relevant pages. Generating irrelevant test questions is simple. We permute the words in a query $\mathcal{Q}$ (e.g. "ice age 2") together with 3 random words to form $\mathcal{Q}'$ (e.g. "company cook 2 over age ice"). To create the test question, we present $\mathcal{Q}$ with the snippet for $\mathcal{Q}'$ [Figure 2]. Our implementation yielded snippets that often contain $\mathcal{Q}$, but are also not highly relevant to $\mathcal{Q}$. We pulled the random words from a list of basic English words[2], accessed on 11/10/06. These examples used queries from the Google Zeitgeist archive[3], accessed on 11/10/06. We used the Google API[4] to retrieve the link text and snippets on the first result returned. When this method fails to produce a clear answer, the user can still pass and get a bonus.

[2] `http://ogden.basic-english.org/words.html`
[3] `http://www.google.com/press/zeitgeist/archive.html`
[4] `http://code.google.com/apis/soapsearch/`

We can use questions with a large number of votes one way as test questions. We anticipate that results taken from the top of a query's result page will often be highly relevant so we should quickly amass many highly relevant test questions.

4.4 Vote processing algorithm

Our vote processing algorithm takes as input the matrix $C_{\mathcal{Q}}$, and outputs a subranking of the top k items, for a given k. A loop considers each row, removing pages that have more than some threshold $\mathcal{D}$ of votes and where an overwhelming majority vote a page spam, until k pages have made it through without being removed from $R'_{\mathcal{Q}}$. The algorithm has some very attractive features. Its simplicity makes it easy to understand. It is very conservative; if 50% of the voters feel one way and 50% another, it leaves the page in, erring on the side of keeping spam.

4.5 Usage estimates

Von Ahn and Dabbish [18] have found that 5,000 users playing the ESP Game 24 hours a day could label 425,000,000 images in 31 days. This averages out to around 13.7 million images labeled in one day. Without considering the scoring infrastructure, the ESP Game is quite similar to the game we propose. There, instead of choosing between two categories (highly relevant and not relevant), players attempt to match each other on a label for an image. The main difference in architecture is that the ESP Game allows multiple attempts at a match for a single image, whereas we only allow players one attempt to label a query-snippet pair as either relevant or not relevant. Labeling in the case of our game would involve two players matching on either a highly relevant or not relevant vote. Ignoring passing (which also occurs in the ESP Game), even with both users guessing uniformly at random a match would be expected after every two questions. By Lemma 3, which will be proven later, we can expect the actual probability of a match between rational users to exceed 1/2. We assume players will spend approximately the same amount of time answering a question in the ESP Game as they do in our game. Suppose that our game attracts 2,500 users. Then, using our assumptions about match frequencies, we can estimate that the average number of query-snippet pairs labeled over the course of 24 hours would be 3.4 million.

Work such as that by Chellapilla and Chickering [15] showed that there is a dependence between cloaking, a particular technique used in web spam, and the popularity or *monetizability* of a query. In essence, if a site owner can expect to earn money when the site is ranked highly for a query, there was an increased likelihood of cloaking. We will assume this applies for web spam in general, and not just for cloaking spam.

Suppose our goal is to rank the first page of results, containing at most 10 distinct sites, for the 10,000 most profitable or popular queries. Then there are $100,000$ query-snippet pairs we would like our algorithm to evaluate. We would need at least $\mathcal{D}$ matches per query-snippet pair to apply our algorithm from Section 4.4. If $0 < c_1 \ll 1$ is the probability that we show a snippet from a page ranked in the top ten for that query, then in expectation we require $\mathcal{D}/c_1 \cdot 100,000$ questions be shown to players. More concretely, if $\mathcal{D} = 25$ and $c_1 = 1/5$, we would need $12,500,000$ matches from the game, which could be accumulated in less than four days of gameplay from $2,500$ users. If we relax those numbers somewhat, so that $\mathcal{D} = 10$ and $c_1 = 1/3$, then only $3,000,000$ matches would be required, which could be accomplished in less than one day. If we only wanted to classify the top ten results for each of the 100 most popular queries of the previous day (*i.e.*, from Google Hot Trends[5]), with $D = 25$ and $c_1 = 1/5$ we could accomplish this in $25 \cdot 5 \cdot 1000 = 125,000$ questions evaluated or in less than 1 hour, again assuming 3.4 million questions labeled per day and $2,500$ people playing. While this might not be rapid enough to label the top results for around 91 million queries, an amount Google is reputed to receive per day, [20] it would allow targeting of popular and highly monetizable queries at first, and in the space of a day many of the day's queries could have their top results evaluated.

4.6 The fun factor

Our game must be fun to succeed; we believe it will be. We plan to use queries from Google Zeitgeist[6] and Google Hot Trends[7]. Google Zeitgeist served as source of popular queries, which was discontinued by Google; Google Hot Trends is its successor. Players will be exposed to queries that others find interesting and are currently searching for, so we expect our users to enjoy them. When queries are relevant to the snippet, players will have an opportunity to learn trivia. Finally, irrelevant snippets can be amusing depending on how they differ from the query.

Figures 3 and 4 show a series of eight potential game questions, each generated automatically. The first step of generating these questions involved choosing a query using a list taken from Google Hot Trends[8] from dates in early July 2007 at random. In the second step, some questions were randomly selected to be formed using the test question generation process, described in Section 4.3. For questions formed using the test-generation process, the real query and the query presented to the user differ. For all other questions, the query presented to the user and the search engine were the same. The ranking was selected at random from 1 to 1000 (a limitation of the Google SOAP API[9], but biased to appear closer to the beginning of the (lower 500) than to the end). The Google SOAP API was used to retrieve the link text, snippet, and URL of a site, given a query and a requested rank.

In Figure 3, only the presented query, link text and snippet are shown. This is the same information that would be shown to a user playing an online game. Figure 4 contains the same information as Figure 3, with the addition of the real query, the url of the actual page, and ranking within Google. These figures show examples of the type of questions that are possible to ask. Some query-snippet pairs, like those in question 5, are possible examples of spam; the presented query was equivalent to the real query, the result was ranked highly, but the snippet seems

[5] `http://www.google.com/trends/hottrends`
[6] `http://www.google.com/press/zeitgeist/archive.html`
[7] `http://www.google.com/trends/hottrends`
[8] `http://www.google.com/trends/hottrends`
[9] `http://code.google.com/apis/soapsearch/`

1. **QUERY: nancy daus benoit**
   FOROS—LR21 :: VER TEMA - BRITNEY SPEARS
   to-chris-benoit.freepara.info/
   http://georgia-news.freepara.info/ ... http://fayette-county-ga.freepara.info/ http://nancy-daus.freepara.info/ ...

2. **QUERY: sonya walger**
   SONYA WALGER PICTURES, BIOGRAPHY, FILMOGRAPHY, TRAILERS,
   Sonya Walger Pictures, Biography, Filmography, Trailers,

3. **QUERY: unbreakable**
   HELSINKI UNIVERSITY OF TECHNOLOGY DEPARTMENT OF FOREST PRODUCTS ...
   copies, no matter how unbreakable the copy protection technologies are. ..... probable that all users didn't use the same criteria when answering these ...

4. **QUERY: veggie booty snack food**
   ROADIES
   spend good money to come here? Let us just focus on our. capital town for a moment. ..... Organic food is the end product of organic farming, which ...

5. **PRESENTED QUERY: hagar creator browne**
   ACROSS
   50 Hagar creator Browne. 51 More decayed. 53 Hamburger'sarticle. 54 Gold standards. 56 Gland: Prefix. 58 Going according to plan. 60Crystal___ ...

6. **QUERY: ernie harwell**
   RADIO HALL OF FAME - ERNIE HARWELL, SPORTSCASTER
   Ernie Harwell is the long-time voice of the Detroit Tigers. He began his career with the Tigers in 1960 and, with the exception of 1992, when he worked for ...

7. **QUERY: cristina fernandez de kirchner**
   BBC NEWS — SPECIAL REPORTS — THE WORLD THIS WEEK
   Cristina Kirchner. Cristina Fernandez de Kirchner is already a senator ... Cristina Fernandez de Kirchner, wife of Argentina's current head of state, ...

8. **QUERY: society of vacuum coaters scholarship**
   IAU COMMISSION 46 ONLINE NEWSLETTERS
   At this stage, small but significant steps may be taken to support this .... The UK Royal Astronomical Society's annual National Astronomy Meeting in 2004 ...

Figure 3: A series of potential questions that players could be asked to judge as highly relevant or not highly relevant.

1. **PRESENTED QUERY: nancy daus benoit**
   FOROS—LR21 :: VER TEMA - BRITNEY SPEARS
   to-chris-benoit.freepara.info/ http://georgia-news.freepara.info/... http://fayette-county-ga.freepara.info/ http://nancy-daus.freepara.info/ ...
   REAL QUERY: nancy daus benoit
   URL: http://foros.lr21.com/viewtopic.php?t=7499&start=510&sid=53a6e780868b53254de55f090952fd79
   Rank: 742

2. **PRESENTED QUERY: sonya walger**
   SONYA WALGER PICTURES, BIOGRAPHY, FILMOGRAPHY, TRAILERS,
   Sonya Walger Pictures, Biography, Filmography, Trailers,
   REAL QUERY: sonya walger
   URL: http://www.starpulse.com/Actresses/Walger,_Sonya/index.html
   Rank: 3

3. **PRESENTED QUERY: unbreakable**
   HELSINKI UNIVERSITY OF TECHNOLOGY DEPARTMENT OF FOREST PRODUCTS ...
   copies, no matter how unbreakable the copy protection technologies are. ..... probable that all users didn't use the same criteria when answering these ...
   REAL QUERY: liquid probable unbreakable little
   URL: http://www.media.hut.fi/~julkaisut/diplomityot/DI_K_Pietila_2005.pdf
   , Rank: 463

4. **PRESENTED QUERY: veggie booty snack food**
   ROADIES
   spend good money to come here? Let us just focus on our. capital town for a moment. ..... Organic food is the end product of organic farming, which ...
   REAL QUERY: vessel veggie food booty snack bright come
   URL: http://www.thdl.org/texts/reprints/midweek/Midweek_01_04.pdf
   Rank: 281

5. **PRESENTED QUERY: hagar creator browne**
   ACROSS
   50 Hagar creator Browne. 51 More decayed. 53 Hamburger'sarticle. 54 Gold standards. 56 Gland: Prefix. 58 Going according to plan. 60Crystal___ ...
   REAL QUERY: hagar creator browne
   URL: http://www.pressofatlanticcity.com/life/story/7489817p−7385293c.html
   Rank: 8

6. **PRESENTED QUERY: ernie harwell**
   RADIO HALL OF FAME - ERNIE HARWELL, SPORTSCASTER
   Ernie Harwell is the long-time voice of the Detroit Tigers. He began his career with the Tigers in 1960 and, with the exception of 1992, when he worked for ...
   REAL QUERY: ernie harwell
   URL: http://www.radiohof.org/sportscasters/ernieharwell.html
   Rank: 3

7. **PRESENTED QUERY: cristina fernandez de kirchner**
   BBC NEWS — SPECIAL REPORTS — THE WORLD THIS WEEK
   Cristina Kirchner. Cristina Fernandez de Kirchner is already a senator ... Cristina Fernandez de Kirchner, wife of Argentina's current head of state, ...
   REAL QUERY: cristina fernandez de kirchner
   URL: http://news.bbc.co.uk/2/hi/in_depth/6897830.stm
   Rank: 5

8. **PRESENTED QUERY: society of vacuum coaters scholarship**
   IAU COMMISSION 46 ONLINE NEWSLETTERS
   At this stage, small but significant steps may be taken to support this .... The UK Royal Astronomical Society's annual National Astronomy Meeting in 2004 ...
   REAL QUERY: living coaters society stage get vacuum of scholarship
   URL: http://physics.open.ac.uk/IAU46/newsletter61.html
   Rank: 130

Figure 4: Questions from Figure 3, combined with the true query, URL, and ranking in Google's index. All information was retrieved using the Google SOAP API (http://code.google.com/apis/soapsearch/). Question 5 was retrieved on July 9, 2007, all others on July 27, 2007.

rather vague and appears to belong to a crossword puzzle. We believe that it is not only possible for people to classify many queries, but that it is also fun. A demo of our game is currently online at `http://www.cs.cmu.edu/~mgoodste/research/demo.html`, which gives a flavor for the full game. No labeling data is currently being collected; matches in the demo are against the authors.

5. STRATEGY-PROOFNESS

In this section we analyze the game and show that the players whose goal is to maximize their score give us their honest opinions. We borrow terminology from game theory and mechanism design. A **strategy** represents the plan a user has for making choices in any possible game situation. A **dominant strategy** is a strategy that maximizes the utility (here, the user's score) when the strategies of other players within the game are unknown. A game is **strategy-proof** if the dominant strategy is to play honestly. For our analysis, we will also define a user's **confidence** as the probability a user assigns to matching an omniscient, honest partner. Throughout this paper, we will assume that a user's confidence and the true probability of a match are approximately equal, and will use these interchangeably.

5.1 The game with no bonus

We begin with a motivating example: Suppose that two users are paired. Player 2 is omniscient and honest. The bonus $\mathcal{B}$ is set to 0. We restrict our analysis to Player 1's actions, since these determine the score for both players. Let p be the confidence Player 1 has in his answer. The expected score on any question is $p(1) - (1-p)(1+\varepsilon)$. By linearity of expectation, the expected score on the entire game is positive if and only if $p > \frac{(1+\varepsilon)}{(2+\varepsilon)}$. Since by passing one can achieve a zero score, answering honestly has a better expected score.

DEFINITION 5.1. *For a given $\varepsilon > 0$, define $p_c = \frac{(1+\varepsilon)}{(2+\varepsilon)}$ as the **threshold confidence** for ε.*

Notice that $p_c > 1/2$, since $\varepsilon > 0$. Also, W.L.O.G. $p \geq 1/2$, since otherwise the confidence in the other answer is $\geq 1/2$. Answering honestly on all questions for which $p > p_c$ and passing otherwise yields a positive expectation for the entire game. No strategy performs better, as it involves more passing or guessing on questions where $p \leq p_c$. By definition, any strategy involving answering with $p \leq p_c$ will have expectation ≤ 0.

Note that once a choice of ε is made, p_c is fixed. By choosing ε, we can alter the confidence above which we wish Player 1 to answer. Our goal is for the dominant strategy for a player to be to play honestly, *irrespective* of the strategy of the opponent. To ensure this, we impose bounds on the bonus $\mathcal{B}$ and penalty ε.

5.2 The game with a bonus

We now consider the game with a bonus, redefining $\mathcal{B}$ as a function of the number of test questions answered correctly when none are answered incorrectly. Note that the partner is no longer assumed to be honest and omniscient.

LEMMA 1. *Let p_c be the confidence threshold. Suppose an honest user answers k test questions. Let $\mathcal{B}(k)$ be the bonus for answering k bonus questions correctly and none incorrectly. Then for answering honestly when $p > p_c$ and passing otherwise to be a dominant strategy, it must be the case that $\forall k' \geq k > 0$, $\frac{\mathcal{B}(k)}{\mathcal{B}(k')} \geq (1/2)^{k'-k}$. If $\mathcal{B}(i) = \beta/p_c^i \ \forall i \in \mathcal{Z}^+$ for some fixed $\beta > 0$, then this necessary condition is satisfied.*

PROOF. By contradiction. Fix a player, and consider those k of the test questions that a player knows with confidence $p > p_c$. Suppose, on the remaining questions, the player guesses. Then his expectation is $p^k(1/2)^{k'-k}\mathcal{B}(k')$. On the other hand, the expectation of just answering the k test questions, and passing on the others, is $p^k\mathcal{B}(k)$. If $\frac{\mathcal{B}(k)}{\mathcal{B}(k')} < (1/2)^{k'-k}$ then $p^k\mathcal{B}(k) < p^k(1/2)^{k'-k}\mathcal{B}(k')$. This implies honesty is not a dominant strategy. Contradiction.

Note that if $\mathcal{B}(i) = \beta/p_c^i$, then $\frac{\mathcal{B}(k)}{\mathcal{B}(k')} = p_c^{k'-k} \geq (1/2)^{k'-k}$. □

REMARK 1. *$B(i) = \beta/p_c^i$ implies that a player never increases his expectation of a bonus by guessing randomly on questions where the honest player would pass.*

For the remainder of the paper, our analysis will assume that $B(i) = \beta/p_c^i$. Consider any dishonest strategy for a player. On the questions she answers, she must employ some linear combination of the following strategies: answering questions the honest strategy would pass on, flipping the honest answer, and giving some honest answers. We demonstrate that the expectation of the bonus when employing any combination of these strategies must be less than in the honest case. After that we prove our main theorem.

LEMMA 2. *Suppose that playing honestly, the player answers k test questions, and passes on the rest. Consider a strategy of answering m test questions honestly, switching the answer on n questions, and answering l questions that the honest strategy would skip, $m, n, l \geq 0$ and $m + n \leq k$. Then honesty gives a larger expected number of points due to the bonus, where $\mathcal{B}(i) = \beta/p_c^i \ \forall i \in \mathcal{Z}^+$*

PROOF. Let $\tilde{p}$ be the minimum confidence the dishonest player has, and $\hat{p}$ the maximum confidence the dishonest player has over all questions he answers. Divide the l questions the dishonest player answers that the honest strategy would skip into two categories, such that the dishonest player answers h honestly with maximum confidence $\hat{p}$ and j dishonestly with confidence $1-\tilde{p}$. We obtain the following set of inequalities.

$$
\begin{aligned}
m, n, j, k, l, h &\geq 0 && (1)\\
j + h &= l && (2)\\
k &\geq m + n && (3)\\
k - m &\geq 0 && (4)\\
p &\geq p_c > 1/2 && (5)\\
p_c &> \hat{p} \geq \tilde{p} \geq 1/2 && (6)\\
p_c > 1/2 &> 1 - \tilde{p} \geq 1 - \hat{p} && (7)\\
p_c &\geq 1 - p && (8)
\end{aligned}
$$

From lines 5 and 4 we get

$$
\begin{aligned}
p/p_c &\geq 1 && (9)\\
(p/p_c)^{k-m} &\geq 1 && (10)
\end{aligned}
$$

Combining lines 2, 5, 6, 7, and using the result from above, we get:

$$
\begin{aligned}
(p_c)^{n+j+h} &\geq (1-p)^n \hat{p}^h (1-\tilde{p})^j\\
(p/p_c)^{k-m}(p_c)^{n+l} &\geq (1-p)^n \hat{p}^h (1-\tilde{p})^j
\end{aligned}
$$

We rearrange terms and eventually multiply by β to get the final result.

$$
\begin{aligned}
\frac{p^k p_c^{m+n+l}}{p^m p_c^k} &\geq (1-p)^n \hat{p}^h (1-\tilde{p})^j\\
\frac{p^k \beta}{(p_c)^k} &\geq \frac{p^m (1-p)^n \hat{p}^h (1-\tilde{p})^j \beta}{p_c^{m+n+l}}\\
p^k B(k) &\geq p^m (1-p)^n \hat{p}^l (1-\tilde{p})^j \mathcal{B}(m+n+l)
\end{aligned}
$$

Finally, E[bonus for honest players] $\geq E$[bonus for dishonest players under any strategy]. □

THEOREM 5.1. *Let $\mathcal{B}(i) = \beta/p_c^i$ be the bonus a player receives for answering $i > 0$ test questions, all correctly. Then playing honestly when $p \geq p_c$ and passing otherwise is a dominant strategy.*

PROOF. By linearity of expectation, we can analyze bonus points and points from matching separately. By Lemma 2, the dominant strategy to maximize bonus points is to play honestly. The dominant strategy over the entire game is the strategy that maximizes the sum of points from matching and the bonus. We can adjust β so that this sum is dominated by the bonus points, thus ensuring that the game is strategy-proof. □

The next lemma states that, assuming users play honestly, a match between two honest players within our game is likelier than a random vote. Theorem 5.1 allows us to assume that rational players behave honestly. Therefore, our game produces data that is likelier and more informative than votes cast uniformly at random.

LEMMA 3. *Pr[two honest users match and generate a vote within a game] > Pr[two users match randomly in a game].*

PROOF SKETCH. Let p and q be the confidence of Players 1 and 2 respectively. Without loss of generality, $q = p + \delta$ for some $\delta \geq 0$. Pr[match in game] $= pq + (1-p)(1-q) = (2p^2 - 2p + 1) + \delta(2p - 1)$. It can be verified that Pr[match in game] $> 1/2 =$ Pr[random match]. □

6. ADVERSARIAL PLAYERS

We define three adversaries. Let Sam be a spammer who wishes to move a non-relevant, spam page $\mathcal{P}$ up in the rankings for a query $\mathcal{Q}$. Let Mallory be an adversary who wants to move a relevant, non-spam page $\mathcal{P}'$ down in the rankings for a query $\mathcal{Q}'$. Let Gene be a generic attacker, who is not interested in any specific page but wishes to corrupt the rankings.

6.1 Sam the spammer

Sam's strategy is to get enough users (bots or humans) that agree with him to play the game. Let $\mathcal{M}/(\mathcal{M}+1)$ be the overwhelming majority our algorithm uses, for $\mathcal{M} \geq 9$ (meaning $\mathcal{M}$ times as many people must vote a page spam as not spam). In doing so, Sam must wait for a query to come up that is relevant to his page $\mathcal{P}$. Also, for each question containing his query, Sam must decide how to vote. We assume that the time restriction prevents Sam from finding what page and ranking value the snippet is associated with. Sam can check the page to see if it is his own snippet, and vote it up. However, even if he sees his own page, Sam will encounter several problems. First, if page $\mathcal{P}$ is truly spam, Sam must hope that his opponent is either his agent or an honest player who thinks $\mathcal{P}$ is relevant. Otherwise, the vote would not be counted in our algorithm.

LEMMA 4 (**Partner agreement**). *Let $\mathcal{P}$ be the page Sam wants to raise in the rankings, and let $0 \leq p_s \ll 1/2$ be the fraction of the honest population who believe that $\mathcal{P}$ is not spam. Let p_m be the probability that $\mathcal{P}$ emerges within one game. Then the expected number of games Sam must play to accumulate one vote in the algorithm is $\frac{1}{p_s p_m}$, assuming that Sam does not have enough agents to affect p_s.*

PROOF. This follows by linearity of expectation. The number of people who believe a page is spam or not is independent of whether a snippet of that page occurs within a game, and thus the probability of Sam both encountering someone who he agrees with and encountering a question he cares about is $p_s p_m$, so he must expect to play $\frac{1}{p_s p_m}$ games to get one match. □

Suppose Sam decides to employ agents to help gain matches. As we are matching players at random, Sam needs many agents. Let H be the number of honest users in the game. Even if we assume a uniform distribution for how players are matched, for Sam to even have a $1/2$ probability of matching his own agent, he must introduce $(1/2 - p_s)H$ agents. Since H is a hidden parameter, as long as H is sufficiently large or p_s is sufficiently low, it is difficult for Sam to add enough agents to affect any page's score. For the same reason, Sam cannot tell whether he has enough agents. Assuming that p_s is low is reasonable; otherwise the page would not be identified as spam under our scheme.

Even if Sam introduces many agents and has an effective strategy for some page-query pair, he has to play a huge number of times to get the pair in the game.

LEMMA 5 (**Too many games**). *Let $n_s = \frac{1}{p_s p_m}$ be the expected number of games Sam must play to achieve one match on $\mathcal{P}$, and $\mathcal{M}n_\mathcal{P}$ be the number of current votes for page $\mathcal{P}$ as spam, with $n_\mathcal{P} \in \mathcal{R}^+$. Let $m_\mathcal{P}$ denote the number of users who think that $\mathcal{P}$ is not spam without Sam's votes. Then Sam must expect to need to play $n_s(n_\mathcal{P} - m_\mathcal{P})$ times to affect the algorithm.*

PROOF. Without loss of generality, assume $n_\mathcal{P} \geq 1$, otherwise there are not enough votes to throw out $\mathcal{P}$ anyway. As a preliminary attempt, assuming no one other than Sam believes his page is not spam, he must expect to play $(n_\mathcal{P})n_s$ games by linearity of expectation. Now, remove the assumption that no one other than Sam believes his page is not spam. Let G be a random variable representing the total number of games Sam must play to keep his page within the rankings. For his page not to be removed, he needs:

$$100np < 100(m_\mathcal{P} + G) \Rightarrow n_\mathcal{P} < m_\mathcal{P} + G \Rightarrow G > n_\mathcal{P} - m_\mathcal{P}$$

By linearity of expectation, since it takes n_s games for Sam to expect one match, it takes $n_s(G) > n_s(n_\mathcal{P} - m_\mathcal{P})$ games to amass enough votes to keep his page in the ranking. □

The final problem Sam encounters is that even **recognizing the snippet** will be difficult. We can embed either the snippet, query, or both, within an image (possibly using a CAPTCHA) to make such comparisons hard for computers. If the game uses a separate snippet from a conventional search engine, even recognizing the snippet can be difficult for a bot.

6.2 Mallory the malicious user

Mallory's attack is the opposite of Sam's attack, but she faces very similar problems to Sam.

LEMMA 6 (**Partner agreement**). *Let $\mathcal{P}'$ be the page Mallory is interested in lowering in the ranking, and $0 \leq p'_s << 1/2$ denote the fraction of the honest population who believe that this page is spam. Let p'_m represent the probability that a snippet from $\mathcal{P}'$ constitutes a question within one game. Then the expected number of games Mallory must play to accumulate one vote in the algorithm is $n'_s = \frac{1}{p'_s p'_m}$, assuming she does not have enough agents to affect p'_s.*

PROOF SKETCH. Equivalent to Lemma 4. □

As with Sam, Mallory needs $\geq (1/2 - p'_s)H$ agents to affect the ranking. We assume p'_s is sufficiently low (otherwise the page would already qualify as spam).

LEMMA 7 (**Too many games**). *Let n'_s be the expected number of games Mallory must play in order to achieve one match on $\mathcal{P}'$, and $\mathcal{M}n'_\mathcal{P}$ be the number of current votes for page $\mathcal{P}'$ as spam, with $n'_\mathcal{P} \in \mathcal{R}^+$. Let $m'_\mathcal{P}$ denote the number of users who think that $\mathcal{P}'$ is not spam. Then Mallory must expect to need to play $\mathcal{M}(m'_\mathcal{P} - n'_\mathcal{P})n'_s$ times to affect the algorithm.*

PROOF. We use a similar derivation to the one in Section 6.1. Once more, let G' be a random variable representing the number of games Mallory must play to affect our algorithm. In order for a page to be thrown out, $100n'_\mathcal{P} + G' > 100m'_\mathcal{P} \Rightarrow G' > 100(m'_\mathcal{P} - n'_\mathcal{P})$. By linearity of expectation, Mallory must play $n'_s G' > 100n'_s(m'_\mathcal{P} - n'_\mathcal{P})$ games in order to affect the game. □

Snippet recognition would be hard for Mallory as well.

6.3 Gene the generic attacker

Gene's attack is to corrupt the ranking. For this, Gene would always vote dishonestly, in an attempt to promote web spam pages within the ranking, and to remove legitimate pages from the ranking. With respect to any question, Gene is either a Sam or a Mallory. When Gene tries to vote a page up, he encounters all of Sam's issues; likewise, whenever Gene tries to vote a page down, he encounters all Mallory's problems. By protecting against Sam and Mallory, we also protect against Gene, since Gene simply has more interests.

In addition, since Gene is attempting to disagree with rational players, he will also disagree with test questions. Since we assume humans cannot tell test and non-test questions apart, neither can bots without large advances in natural language processing (NLP). Hence, Gene should have a history of doing very poorly on the test questions. We can adapt our algorithm to discount votes of users with a poor history on test questions.

6.4 Other attacks

Assuming a smaller set of test questions as compared to non-test, an adversary could write a bot that plays games and scrapes the screen as it goes. Any query-snippet pair that repeats would be a test question. Using a non-uniform distribution on the non-test questions, possibly biasing a small set so that repeats do not necessarily indicate test questions, is one way to prevent this. Another is to have questions move into the test set when enough feedback exists to know the "right answer", and have a question leave the test set once shown several times. Finally, we can decree that each user can only see a query-snippet pair once as a test question; subsequent views are non-test. As $\mathcal{B}$ is individual, we can present different test sets to different users. This may make it easier to allow each user to see each query-snippet pair only once as a test, by keeping a precalculated list of "next test" that is combined to form a game.

6.5 Key assumptions

The game's success depends on certain assumptions (some previously mentioned). (1) A user's confidence well approximates the true probability of two players matching. (2) Bots perform poorly at NLP. (3) The snippet represents the page. (4) The (plentiful) test questions are indistinguishable from other questions. (5) We control the query and page. (6) Users are paired at random. (7) Players cannot research the page's URL or its location in the ranking. (8) People enjoy the game.

7. RELEVANCY JUDGMENTS

In information retrieval (IR), a **relevancy judgment** refers to the process of showing a user a query and some information from a document, and asking him or her to label the document as relevant or not relevant [10]. This information is then fed back to the system to improve the retrieval process. These users, who can be referred to as *judges*, are very similar to our concept of honest agents. Our game achieves a very similar goal–we provide a query, a snippet from a web page, and ask users to rate the page as either highly relevant or not. As none of our results were predicated on the data we were labeling being web pages, researchers in IR could construct a similar game, where instead of a web page's snippet they substitute *data normally shown to judges*. This could provide a way of eliciting relevancy judgments that is much less expensive than current methods.

8. CONCLUSIONS AND FURTHER WORK

In this paper we presented a two-player game approach to combating web spam. We showed it is strategy-proof and that the information obtained from it can be used to find spam pages. First of all, the likelihood of an attack by an adversary is quite low. Because of this, the votes we do gather simulate those of honest agents and a simple voting algorithm that compares votes can be used. As we only ask players to vote a binary choice between spam and not spam, we do not encounter any voting paradoxes. In addition, we provided one of the first schemes we are aware of with provable performance guarantees. We have also demonstrated a connection between our work on web spam and a more general problem in information retrieval, that of soliciting relevancy judgments. Further work will focus on completing our implementation (demo online at `http://www.cs.cmu.edu/~mgoodste/research/demo.html`) and testing our game, as well as assessing our success.

9. ACKNOWLEDGMENTS

The authors are extremely grateful to Manuel Blum and Luis von Ahn for their insightful ideas and thoughtful commentary as the work was developed. The authors would also like to thank Anthony Tomasic for the helpful discussion of relevancy judgments and information retrieval.

10. REFERENCES

[1] B. Wu and B. D. Davison. Identifying link spam farm pages. In *WWW*, 2005.

[2] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *WebDB*, 2004.

[3] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *AIRWEB*, 2005.

[4] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB*, 2004.

[5] Z. Gyongyi and H. Garcia-Molina. Link spam alliances. Technical report, Stanford University, 2005.

[6] Z. Gyongyi, H. Garcia-Molina, P. Berkhin, and J. Pedersen. Link spam detection based on mass estimation. In *VLDB*, 2006.

[7] B. Wu, V. Goel, and B. Davidson. Topical trustrank: Using topicality to combat web spam. In *WWW*, 2006.

[8] Y. Wang, M. Ma, Y. Niu, and H. Chen. Spam double-funnel: Connecting web spammers with advertisers. In *WWW*, 2007.

[9] Y. Du, Y. Shi, and X. Zhao. Using spam farm to boost PageRank. *AIRWeb*, 2007.

[10] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[12] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *AIRWEB*, 2006.

[13] A. L. da Costa Carvalho, P.-A. Chirita, E. S. de Moura, P. Calado, and W. Nejdl. Site level noise removal for search engines. In *WWW*, 2006.

[14] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *WWW*, 2006.

[15] K. Chellapilla and D.M. Chickering. Improving Cloaking Detection Using Search Query Popularity and Monetizability. *AIRWeb*, 2006.

[16] J. Caverlee and L. Liu. Countering web spam with credibility-based link analysis. In *PODC*, 2007.

[17] J. Caverlee, S. Webb, and L. Liu. Spam-resilient web rankings via influence throttling. In *IPDPS*, 2007.

[18] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *CHI*, 2004.

[19] A. Pansari and M. Mayer. `http://googleblog.blogspot.com/2006/04/this-is-test-this-is-only-test.html`, 2006. Accessed on 02/03/07.

[20] D. Sullivan. `http://searchenginewatch.com/showPage.html?page=2156461`. Accessed on 8/1/07.