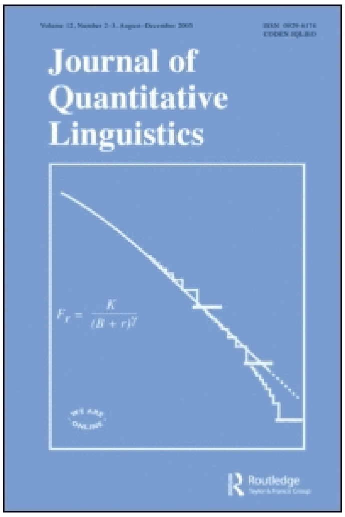


**Journal of Quantitative Linguistics**
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t716100702

**Internet Search Result Probabilities: Heaps' Law and Word Associativity***

Jonathan C. Lansey [ab]; Bruce Bukiet [a]
[a] Department of Mathematical Sciences, New Jersey Institute of Technology, USA [b] Cognitive and Neural Systems Department, Boston University, USA

Online Publication Date: 01 February 2009

**To cite this Article** Lansey, Jonathan C. and Bukiet, Bruce(2009)'Internet Search Result Probabilities: Heaps' Law and Word Associativity*',Journal of Quantitative Linguistics,16:1,40 — 66

**To link to this Article: DOI:** 10.1080/09296170802514153

**URL:** http://dx.doi.org/10.1080/09296170802514153

Journal of Quantitative Linguistics
2009, Volume 16, Number 1, pp. 40–66
DOI: 10.1080/09296170802514153

Routledge
Taylor & Francis Group

# Internet Search Result Probabilities: Heaps' Law and Word Associativity*

Jonathan C. Lansey[1,2] and Bruce Bukiet[1]
[1]Department of Mathematical Sciences, New Jersey Institute of Technology, USA;
[2]Cognitive and Neural Systems Department, Boston University, USA

## ABSTRACT

We study the number of internet search results returned from multi-word queries based on the number of results returned when each word is searched for individually. We derive a model to describe search result values for multi-word queries using the total number of pages indexed by Google and by applying the Zipf power law to the words per page distribution on the internet and Heaps' law for unique word counts. Based on data from 351 word pairs each with exactly one hit when searched for together, and a Zipf law coefficient determined in other studies, we approximate the Heaps' law coefficient for the indexed worldwide web (about 8 billion pages) to be $\beta = 0.52$. Previous studies used under 20,000 pages. We demonstrate through examples how the model can be used to analyse automatically the relatedness of word pairs assigning each a value we call "strength of associativity". We demonstrate the validity of our method with word triplets and through two experiments conducted 8 months apart. We then use our model to compare the index sizes of competing search giants Yahoo and Google.

## INTRODUCTION

With the growth of the internet and the worldwide web there has been a paradigm shift in the way that people approach researching topics. Whereas in the previous generation, the first place to look was the card

*Address correspondence to: Jonathan Lansey, 677 Beacon Street, Boston, MA 02215, USA. Tel: (973) 885-2893. Fax: (973) 596-5591. E-mail: jcl7@njit.edu catalogue at a library, or in an encyclopaedia, nowadays people are "Googling" some keywords.

Much research has been performed concerning efficient ways for search engines to crawl, index and rank web pages (Brin & Page, 1998). However, in this paper, we are interested in understanding the number of search results returned by sets of words.

We begin by developing a basic model to predict the expected value and probability distribution for the number of results returned when a pair of unrelated words is entered into the Google search engine. We then develop a more realistic model by taking into account large variation in page size on the internet. The model requires the distribution for the number of unique words on a web page which we calculate by combining the Zipf's law (Zipf, 1932; Adamic & Huberman, 2002) for the distribution of website text sizes (i.e. the number of words on a webpage), and Heaps' law (Heaps, 1978; Beaza-Yates & Ribeiro-Neto, 1999) for vocabulary size in a document of a given length.

To calibrate and test the model, we use data from 351 word pairs that return exactly one result when submitted to Google (called Googlewhacks, and taken from the website www.googlewhack.com). We also have employed a computer program (Bukiet, 2005) that automatically submits the word pairs to Google and returns the number of results for each word individually and as a pair

Calibrating the model with Googlewhacks allows us to measure the coefficient $\beta = 0.52$ for Heaps' law. Previous studies using the internet have used text collections with under 20,000 pages (Heaps, 1978; Beaza-Yates & Ribeiro-Neto, 1999). In this paper, we demonstrate a powerful method for measuring Heaps' law quickly for very large libraries; in our case, approximately 8 billion pages.

We investigate the association of various word pairs, determining a value we call "strength of associativity" for each pair. We also extend the model to consider searches with more than two words using the same Zipf's law and Heaps' law parameters determined from pairs, and provide a sample of results for groups of three words. Our formula for the expected number of search results includes a parameter for the current number of pages indexed by the search engine. Data was collected with two experiments, 8 months apart, over which this "index size" parameter approximately tripled. The resulting plots are described well by the model, further demonstrating the validity of our method. We then compare the index sizes of two competing search giants, Yahoo and

Google. Finally we discuss the extent of and reasons for differences between theoretical and experimental results.

## BASIC MODEL (UNIFORM)

### Finding the Expected Number of Hits

Let $I =$ the number of web pages indexed by Google. If a given word has $A$ results when searched alone, assuming all pages indexed by Google to be alike, the probability of the word appearing on any given page is $A/I$. The probability for a second word to appear on a page is likewise $B/I$, with $B$ equalling the number of results when the second word is searched for alone. If independence can be assumed, then the probability of a given page having both of the words will be the product of the individual probabilities

$$\frac{A}{I}\frac{B}{I} = \frac{AB}{I^2}.$$

If we wish to ascertain the expected *number* of results when searching through the entire index, we multiply the probability for a single page by the number of pages in the index:

$$I\frac{AB}{I^2} = \frac{AB}{I} = R = \text{Expected number of results}$$

$$AB\frac{1}{I} = R. \tag{1}$$

By “expected results” we mean the average number of results when many words with individual results $A$, and $B$ are searched for.

### Finding the Distribution of Hits

We have so far used the expected results as our theoretical measure, but to get some quantitative insight into the error, we need to find the probability distribution of results. Under the assumption that all pages are equal, we can calculate the probability distribution using combinatorics. Let $p(R)$ be the probability of there being exactly $R$ pages that have both the first and second word on the page. To find this probability, we first find the total number of ways to arrange the $A$ pages with the first word on the $I$ pages in our "internet,"

$$\binom{I}{A} = \frac{I!}{A!(I-A)!}$$

and set this as our sample space. We then find the number of cases where there are exactly $R$ hits from among all these different arrangements. This is equal to a product of two factors. The first factor is the number of ways to arrange $R$ hits among the $B$ pages, or

$$\binom{B}{R} = \frac{B!}{R!(B-R)!}.$$

The second factor is the number of different ways to arrange all the leftover hits from $A$ that were not in $R$ or $(A - R)$. These leftover hits are not on any of the $B$ pages because the pages where $A$ and $B$ coincide are defined as a hit, or part of $R$, and so they can be arranged over $(I - B)$ pages with the number of total arrangements being

$$\binom{I-B}{A-R} = \frac{(I-B)!}{(A-R)!(I-B-A+R)!}.$$

The product of these two factors divided by the size of the sample space (the first number) will give us the probability of a word pair returning $R$ results:

$$\begin{aligned} P(R) &= \binom{B}{R}\binom{I-B}{A-R}\Big/\binom{I}{A} \\ p(R) &= \frac{B!}{R!(B-R)!}\frac{(I-B)!}{(A-R)!(I-B-A+R)!}\frac{A!(I-A)!}{I!} \qquad (2) \\ p(R) &= \frac{A!B!(I-B)!(I-A)!}{I!R!(A-R)!(B-R)!(I-A-B+R)!}. \end{aligned}$$

This is known as the hypergeometric distribution and is the same formula used both by Ziegler (2002) and Altmann (1988) in their own linguistic studies. It can be shown (Weissstein, 1999) that its sum over all valid values of $R$ is equal to 1, and that the expectation of this distribution over all $R$ gives Equation (1),

$$AB\frac{1}{I} = R$$

which confirms our earlier solution for the expected number of results.

Unfortunately these numbers grow large very quickly as $A$, $B$, $I$ or $R$ grow. The formula is impractical even for values of $I > 200$ while we would like to scale $I$ to be on the order of 8 billion. We used Stirling's approximation (Feller, 1968) to calculate $p(R), n! \approx \sqrt{2\pi}e^{-n}n^{n+1/2}$. But since the sheer size of the numbers is the problem, only the log of the Stirling's approximation was practical for us to use:

$$\log(n!) \approx \frac{1}{2}\log(2\pi) - n + (n + 1/2)\log(n).$$

Thus, the probability of having $R$ pages returned when searching for the first and second words in an internet of $I$ pages is:

$$\begin{aligned} p(A, B, I) = \exp(&(A + 1/2) \cdot \log(A) + (B + 1/2) \cdot \log(B) \\ &+ (I - A + 1/2) \cdot \log(I - A) + (I - B + 1/2) \cdot \log(I - B) \\ &- (I + 1/2) \cdot \log(I) - (R + 1/2) \cdot \log(R) - (A - R + 1/2) \\ &\cdot \log(A - R) - (B - R + 1/2) \cdot \log(B - R) \\ &- (I + R - A - B + 1/2) \cdot \log(I + R - A - B) \\ &- (1/2) \cdot \log(2 \cdot \pi)). \end{aligned}$$

We will return to this formula in the discussion when we consider the errors of the more detailed model.

## DETAILED MODEL (NON-UNIFORM)

### Model Assumptions Taking into Account Variation of Page Sizes

It seems most reasonable to question our initial assumption that all pages are equal. Since some pages have more words than others, the probability of finding a result on a given page is not the same for each page. More results will be returned because words are more likely to be found together on the same (lengthier) pages than on shorter pages. This larger number of results for word pairs should lead to combinations of rarer words giving Googlewhacks.

If we are to take different page sizes into account, we need to know the distribution of page sizes on the web. More precisely, since the probability that a word will be found on a page is proportional to number of unique words on the page, we need the distribution of unique words per page for the pages of Google's index.

## Zipf's Law

A number of studies have shown that many different internet phenomena, such as file sizes, follow a power law called the Zipf law, described in detail below (Zipf, 1932; Falutsos, M. et al., 1999). Although the number of words on a page does not necessarily equate to the size of the html file it is encoded in, and some of the studies included other media as well as text, it is reasonable to assume that the number of words per page distribution follows a Zipf law. Theoretical justification for Zipf's law, under assumptions also valid for number of words per web page, was provided by Huberman (Adamic & Huberman, 2002) and Downey (2001).

The Zipf law states that the size of the *n*th largest entry, in our case the *n*th largest web page, is proportional to $1/n^{\alpha}$ where often $\alpha \approx 1$. As stated earlier, there are $I$ pages in Google's index. After ranking the pages in order from the most words to the least words $[1 \cdots i \cdots I]$ let $i$ be the $i$th number in that set. Applying the Zipf law:

$$[\#Words/Page](i) \propto \frac{1}{i^{\alpha}} \text{ where } \alpha \approx 1. \tag{3}$$

## Heaps' Law

We use Heaps' law to convert total words per page into the number of *unique* words per page. When $V_R(n)$ is the portion of the vocabulary $V(V_R \subseteq V)$ represented by the text of size $n$, i.e. $V_R(n)$ is the number of unique words in a text with $n$ total words, and $\beta$ is a free parameter determined empirically, Heaps' law takes the form shown by Equation (4):

$$V_R(n) \propto n^{\beta}. \tag{4}$$

Merging Equations (3) and (4) gives Equation (5) for the number of *unique* words:

$$\text{Total unique words on the } i\text{th largest page } \propto n^{\beta} \propto \left(\frac{1}{i^{\alpha}}\right)^{\beta} \propto \frac{1}{i^{\alpha\beta}}. \quad (5)$$

**Derivation of Model**

Suppose for a particular word there is a single hit. The probability of this word appearing on page $i$ will be called $p(i)$. Since this is proportional to the number of unique words on page $i$, we multiply Equation (5) by a constant $k$ to obtain Equation (6) for the probability of a hit on this $i$th page.

$$p(i) = \frac{k}{i^{\alpha\beta}} \text{ for } i \in (1 \cdots I) \quad (6)$$

Since there is only one hit, the probabilities must sum to one:

$$\sum\nolimits_{1}^{I} p(i) = 1 = \sum\nolimits_{1}^{I} \frac{k}{i^{\alpha\beta}}. \quad (7)$$

For large $I$, we can approximate this sum by an integral,

$$k \int_{1}^{I} \frac{1}{i^{\alpha\beta}} di = 1.$$

From this we can find a formula for $k$ in terms of $\alpha\beta$ and $I$:

$$1 = k \int_{1}^{I} \frac{1}{i^{\alpha\beta}} di = k \left[ \frac{I^{1-\alpha\beta}}{1-\alpha\beta} - \frac{1^{1-\alpha\beta}}{1-\alpha\beta} \right] = k \frac{I^{1-\alpha\beta} - 1}{1-\alpha\beta}$$

$$k = 1 \div \frac{I^{1-\alpha\beta} - 1}{1-\alpha\beta} = \frac{1-\alpha\beta}{I^{1-\alpha\beta} - 1}.$$

Plugging this formula for $k$ into Equation (6) gives Equation (8):

$$p(i) = \left( \frac{1-\alpha\beta}{I^{1-\alpha\beta} - 1} \right) \frac{1}{i^{\alpha\beta}}. \quad (8)$$

This is an important formula and we will come back to it when we derive a formula for word triplets. For an index of a million pages, our integral approximation to find

$$k = \frac{1-\alpha\beta}{I^{1-\alpha\beta}-1}$$

is reasonable; the exact sum of the resulting probabilities is 1.0003.

But how does this probability change when more than one hit occurs in the index? In other words, if a word has $A > 1$ results, what is $p(A,i)$. While it is possible to work this through analytically, it is quite messy and quickly becomes impractical for moderately large values of $I$. We will make an approximation that $p(A, i) \approx A \cdot p(1, i)$ for $A \ll I$. This result makes intuitive sense. In a two-page index, adding a second hit will greatly increase probabilities, but one more hit in a vast index will have little effect on the probability that one of the hits will occur on a particular page. The chances of the second hit landing on a given page $i$ will be quite close to $p(i)$, yielding $p(2, i) \approx 2p(1, i)$. To test this approximation, we ran a computational model for an index of 100 pages and a probability distribution of

$$p(i) = \frac{k}{i^{1/2}}.$$

Each curve in Figure 1 represents a given page in the index. The $y$ axis represents the probability of a word being found on a particular page (largest, fourth-largest, etc.) and the $x$ axis represents the number of hits for the word. The dashed straight lines represent the linear approximations for the probabilities of each of the pages. Since the majority of words we are interested in return orders of magnitude fewer pages than $I$, this approximation should be sufficiently accurate for our purposes.

Now we can ask what is the probability of two different words, which return $A$ and $B$ results respectively, occurring together on the $i$th largest page. To find this, we multiply the two probabilities.

$$Ap(i) \cdot Bp(i) = AB \cdot p^2(i) = AB \cdot \left(\frac{1-\alpha\beta}{I^{1-\alpha\beta}-1}\right)^2 \frac{1}{i^{2\alpha\beta}}.$$

To find the expected number of results for the whole index, we again take a sum and approximate it with an integral. With an index of one

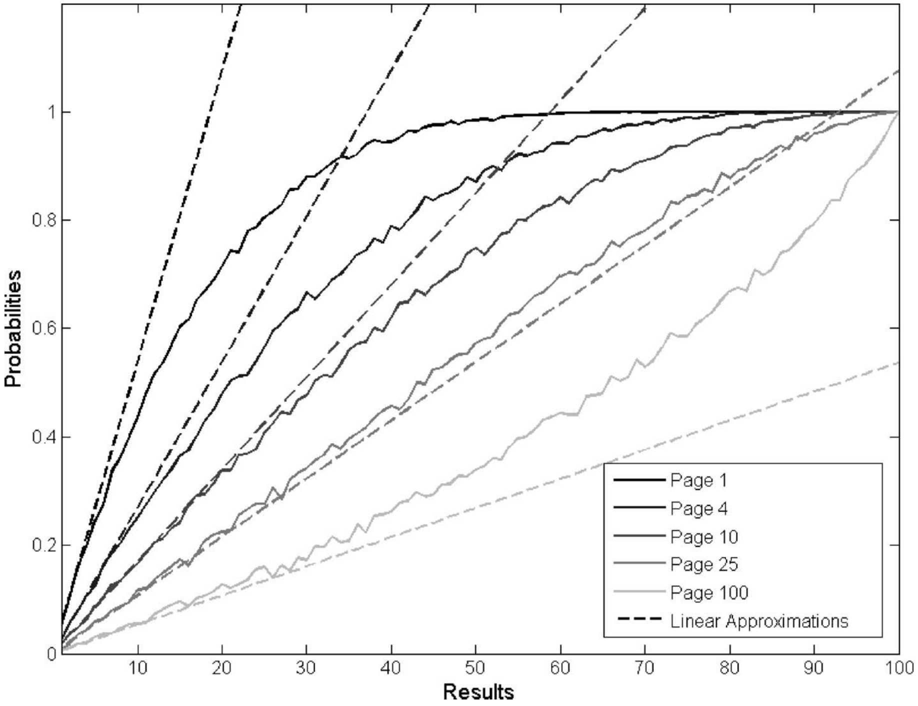


Fig. 1. Probability of a word being found on a given page versus the total number of pages that word is on (Results). Page 1 is the largest and the last page, 100, is the smallest. This computational experiment verifies that the linear approximation discussed in the text is reasonable for small values of hits/index size (I).

million, this integral approximation underestimates results by 5%, the error goes down as the index becomes larger:

$$\int_1^I AB\cdot\left(\frac{1-\alpha\beta}{I^{1-\alpha\beta}-1}\right)^2\frac{1}{i^{2\alpha\beta}} = AB\cdot\left(\frac{1-\alpha\beta}{I^{1-\alpha\beta}-1}\right)^2\int_1^I\frac{1}{i^{2\alpha\beta}} = \left(\frac{1-\alpha\beta}{I^{1-\alpha\beta}-1}\right)^2 \left(\frac{I^{1-2\alpha\beta}-1}{1-2\alpha\beta}\right) = AB\cdot\frac{(\alpha\beta-1)^2\left(I^{2\alpha\beta}-I\right)}{(2\alpha\beta-1)(I^{\alpha\beta}-I)^2} = R$$
$$= \text{Expected number of results.} \qquad (9)$$

As a check for the new formula, we see that in the limiting case of large $I$, and when the Zipf's law exponent $\alpha=0$ (a uniform distribution)

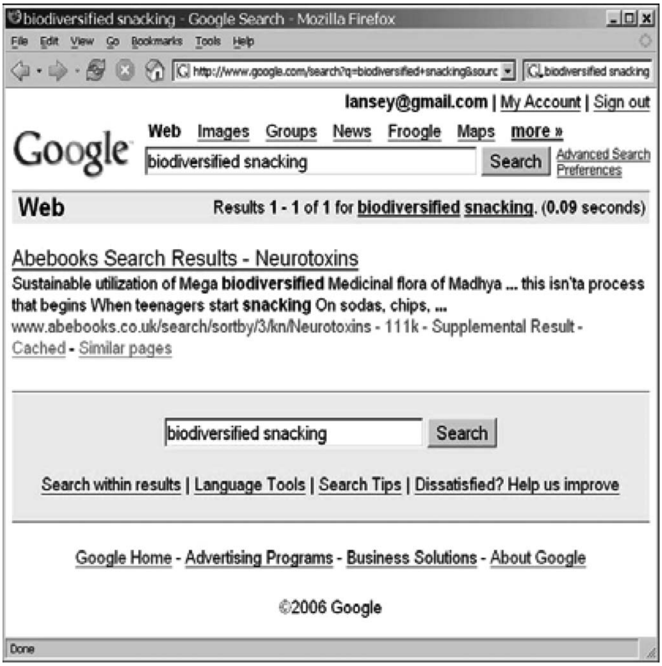
Equation (9) reduces to Equation (1) derived in the "Basic (Uniform) Model" section.

$$\lim_{I\to\infty}\left[AB\cdot\frac{(0-1)^2\left(I^0-I\right)}{(0-1)\left(I^0-I\right)^2}\right]=\lim_{I\to\infty}\left[AB\cdot\frac{(1-I)}{-(1-I)^2}\right]=AB\cdot\frac{1}{I}.$$

## Introduction to Googlewhack

To test the model, we introduce an internet game called Googlewhacking (see, www.googlewhack.com) in which the player submits two English words into the Google search engine attempting to find exactly one hit. Biodiversified Snacking, for example, as shown in Figure 2. Table 1 has a few more select examples of word pairs that were once Googlewhacks.

If the player succeeds, he or she may then submit the word to the Googlewhack website where the word pair is posted on the "Whack Stack", thereby leading to the requisite 15 minutes of fame for the



biodiversified snacking - Google Search - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.google.com/search?q=biodiversified+snacking&sourc
biodiversified snacking
lansey@gmail.com | My Account | Sign out
Google
Web Images Groups News Froogle Maps more »
biodiversified snacking
Search
Advanced Search
Preferences
Web

Results **1** - **1** of **1** for **biodiversified snacking**. (**0.09** seconds)

Abebooks Search Results - Neurotoxins

Sustainable utilization of Mega **biodiversified** Medicinal flora of Madhya ... this isn'ta process that begins When teenagers start **snacking** On sodas, chips, ...

www.abebooks.co.uk/search/sortby/3/kn/Neurotoxins - 111k - Supplemental Result - Cached - Similar pages

biodiversified snacking
Search
Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve
Google Home - Advertising Programs - Business Solutions - About Google
©2006 Google
Done

Fig. 2. An example of a Googlewhack.

Table 1. Six examples of old Googlewhacks.

| |
|---|
| Biodiversified Snacking |
| Fabulated Marshmellows |
| Protozoic Spliff |
| Slipperiest Airscrew |
| Quintupling Zugzwang |
| Netherworldly Mugwumps |

successful player. While simply entertainment for most people the ‘‘Whack Stack’’ provides a unique opportunity to fix our variable for the number of results. This allows us to easily calibrate the model by choosing a value of $\alpha\beta$ that best fits the Googlewhack data. For the numerical part of the experiment, 500 word pairs were taken from the whack stack and those that were no longer Googlewhacks weeded out. 351 of the pairs still returned one result and these were used in the following data analysis.

### Results from Googlewhacking

In this section we use Googlewhack results and Equation (9) to approximate the Zipf and Heaps' parameter product, $\alpha\beta$.

Equation (9) implies that the product $AB$ should remain constant if $R$ equals a constant; in our case of Googlewhacks, $R$ is one:

$$AB \cdot \frac{(\alpha\beta - 1)^2 \left(I^{2\alpha\beta} - I\right)}{(2\alpha\beta - 1)(I^{\alpha\beta} - I)^2} = 1.$$

The values of $A \cdot B$ for the 351 Googlewhacks (which can be found at http://web.njit.edu/∼jcl7/publications/googlewhack.html) formed an approximate lognormal distribution with a nice peak shown by the histogram in Figure 3. The solid line is a lognormal distribution with $\sigma = 2.01$ and $\mu = \log(932{,}260{,}000)$. We note that tests with searching for random numbers show that internet quantity of search results returned for single number queries generally follow an approximately lognormal distribution. This pattern suggests we should take a geometric mean of the data in this histogram, or an average of the logarithms of these data which will be near the peak of the fitted lognormal curve at $\mu$. We note that this solution does not have the status of an adequate model. It would be worthwhile to determine parameters and set up a proper model for this in the future.

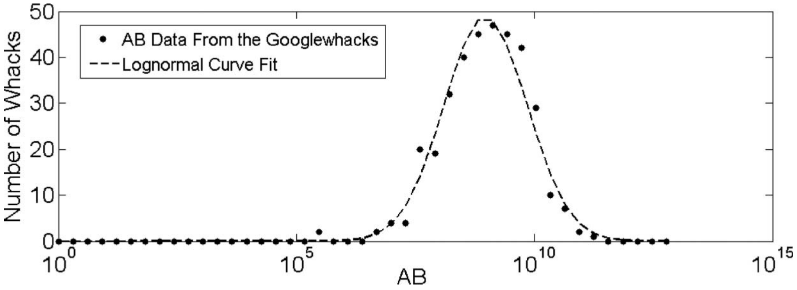


Fig. 3. This figure plots a lognormal distribution fitted to a histogram of the all values of *AB* for the Googlewhacks used in the study.

The geometric mean of $A \cdot B$ for the 351 Googlewhacks was 932,260,000, which we compare to the uniform distribution's theoretical measure (actually, what Google cited as its number of pages indexed at that time) for this value of 8,058,044,651. We note that 75% of the points on the curve lie exactly on some lognormal distribution for $1.93 < \sigma < 2.10$ and $\log(823{,}900{,}000) < \mu < \log(1{,}054{,}900{,}000)$.

Going back to Equation (9), we can plug in $I = 8{,}058{,}044{,}651$ from the Google homepage and we can plug in $AB = 932{,}260{,}000$ from the geometric mean of $AB$:

$$AB \cdot \frac{(\alpha\beta - 1)^2 \left(I^{2\alpha\beta} - I\right)}{(2\alpha\beta - 1)\left(I^{\alpha\beta} - I\right)^2} = 932,260,000 \frac{(\alpha\beta - 1)^2 \left(I^{2\alpha\beta} - I\right)}{(2\alpha\beta - 1)\left(I^{\alpha\beta} - I\right)^2} = 1.$$

We can find $\alpha\beta$ with a computer by approximating the zero of the following equation:

$$932,260,000 \frac{(\alpha\beta - 1)^2 \left(I^{2\alpha\beta} - I\right)}{(2\alpha\beta - 1)\left(I^{\alpha\beta} - I\right)^2} - 1 = 0. \tag{10}$$

We find that $\alpha\beta = 0.520$, or using the range of 75% accuracy (i.e. varying *AB* from 823,900,000 to 1,054,900,000), we find that $0.514 < \alpha\beta < 0.526$.

This result validates our assumptions because Adamic and Huberman (2002) measured the Zipf law coefficient for their studies of the internet to be $\alpha \approx 1$, putting $\beta$ nicely within the bounds $0.4 \leq \beta \leq 0.6$ determined by Baeza-Yates (Baeza-Yates & Ribeiro-Neto, 1999). If either the Zipf's or

Heaps' law parameters are measured more accurately for internet pages independently, a more accurate measure for the other can be easily calculated. In our study, we assume the Zipf law coefficient to be unity, making this the largest study of Heaps' law ever undertaken. Previous studies using the internet have used text collections with under 20,000 pages – about 5 orders of magnitude less than the 8,000,000,000 pages used here (Heaps, 1978; Baeza-Yates & Ribeiro-Neto, 1999; French, 2002).

## Effective Index Sizes

We can get a bit more intuition for Equation (9) when we compare it to our results from the basic (uniform) model, Equation (1):

$$AB\cdot\frac{1}{I}=R,\ AB\cdot\frac{(\alpha\beta-1)^2\left(I^{2\alpha\beta}-I\right)}{(2\alpha\beta-1)(I^{\alpha\beta}-I)^2}=R.$$

The formulas differ only by a constant. We define a new constant to be:

$$\frac{1}{I_{eff2}}=\frac{(\alpha\beta-1)^2\left(I^{2\alpha\beta}-I\right)}{(2\alpha\beta-1)(I^{\alpha\beta}-I)^2}\text{ so that }\frac{AB}{I_{eff2}}=R. \tag{11}$$

This formula suggests an intuitive description of $I_{eff2}$ as the effective index size, the subscript 2 means it only applies to pairs of words. We will derive and use $I_{eff3}$ later.

Set $\alpha=1$ and $\beta=0.52$:

$$\frac{(\alpha\beta-1)^2\left(I^{2\alpha\beta}-I\right)}{(2\alpha\beta-1)(I^{\alpha\beta}-I)^2}=\frac{1}{I_{eff2}}=\frac{1}{932,260,000}. \tag{12}$$

## Googlewhack Plots

Since a Googlewhack has exactly one result, to test the theory we set the expected number of results, $R$, to equal one and then plug in an empirical value for $I_{eff2}$.

We make a log-log plot of the results for each word in a pair, $A$ and $B$, versus their ratio, $A/B$. That is, we plot $\log(A)$ versus $\log(A/B)$ and $\log(B)$

versus $\log(A/B)$. With this approach the points, in theory, should lie on two lines, demonstrated as follows:

Since $AB = I$

$$\log(I) = \log(AB) = \log(A) + \log(B)$$

$$\log(A) = \log(I) - \log(B).$$

Let

$$x = \log(A/B) = \log(A) - \log(B) = \log(I) - 2\log(B)$$

$$y1 = \log(B) = (\log(I) - x)/2 = \log\sqrt{I} - x/2$$
$$y2 = \log(A) = x + \log(B) = \log\sqrt{I} - x/2 + x = \log\sqrt{I} + x/2. \quad (13)$$

The results for the 351 Googlewhack pairs are plotted in Figure 4 along with the theoretical lines where these points should lie based on Equation (13).

## Application of the Model to Determining Relatedness of Words

Since

$$\frac{AB}{I_{eff2}} = R = \text{expected number of results},$$

we are not only limited to Googlewhacks. We expect that word pairs in which the words are closely related will show many more results than expected from the random process of our model. Eight such pairs were tested including the following queries: ‘‘Stairway Heaven’’, ‘‘Train Station’’ and ‘‘Britney Spears’’.

Six hundred and ninety six other pairs were tested consisting of combinations taken from the Googlewhack vocabulary.

The results are plotted in Figure 5. Each point represents data from a word pair and is positioned where $(x,y) = (AB,R)$. As usual, a log-log plot is used. The formula for the solid line comes directly from Equation (11),

$$\frac{AB}{I_{eff2}} = R$$

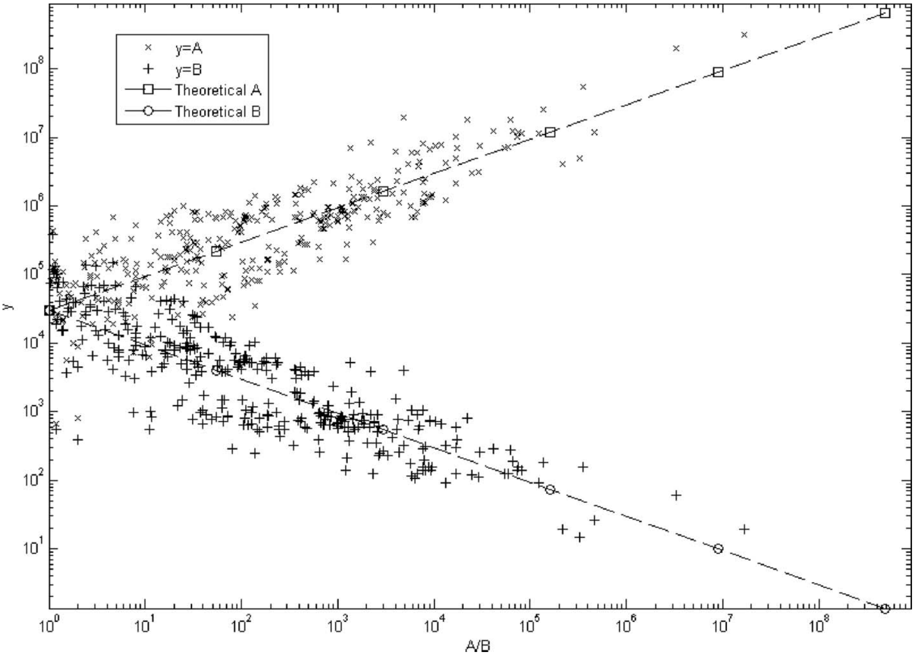


Fig. 4. A log-log plot of the number of hits for the first and second words that make up Googlewhack pairs versus the log of the ratio of hits $A/B$. The lines represent where the × ($A$) and + ($B$) should lie theoretically for the Uniform model.

$$\log(R) = \log(AB) - \log\left(I_{eff2}\right)$$

or

$$y = x - \log\left(I_{eff2}\right). \tag{14}$$

The points marked with circles are the "associated" words. That is, these words are in some sense more related than randomly-chosen words. Although they all lie above the line as they should (actual > expected), they are not far away from the regular error. We define the "strength of associativity" of a pair of words as log of the quotient of the actual number of results and the expected number of results:

$$SA = \log\left(\frac{R_{\text{actual}}}{\mathrm{R}_{\text{expected}}}\right).$$

The eight pairs and their corresponding values for *SA* are shown in Table 2.

Another surprising result of the graph is the asymmetric distribution of points around the theoretical line from Equation (14), with clearly more points towards the left of this line. This asymmetry is more pronounced

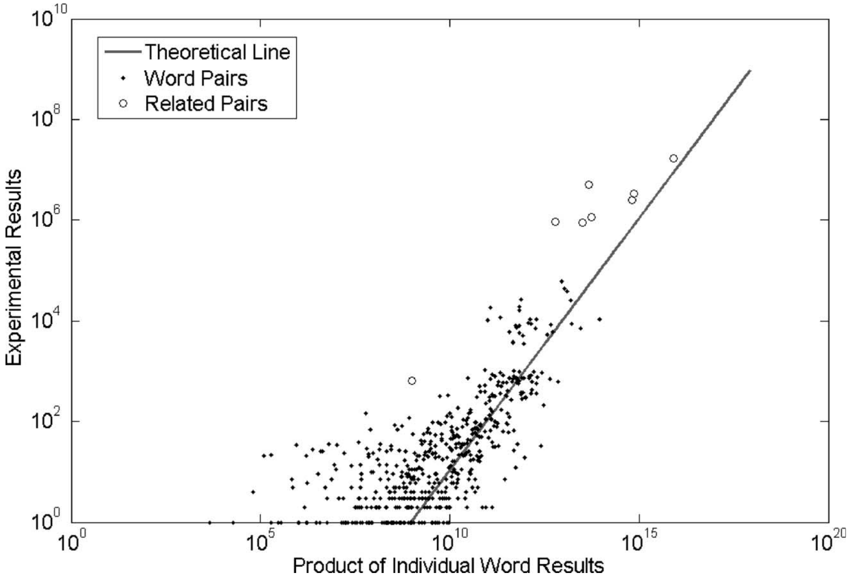


Fig. 5. This figure plots the results for both words together versus the product of the individual results. The solid line is the theoretical expectation of *AB* for a given result *R*.

Table 2. Hand-picked associated word pairs along with the number of results expected, actual results returned by Google and the corresponding value for "strength of associativity".

| Word pair | Actual results | Expected results | Strength of associativity |
|---|---|---|---|
| Paintable Paintability | 635 | 1 | 2.780 |
| Smashing Pumpkins | 930,000 | 6442 | 2.159 |
| Britney Spears | 5,130,000 | 49,016 | 2.020 |
| Stairway Heaven | 893,000 | 35,151 | 1.405 |
| Surge Protector | 1,150,000 | 59,259 | 1.288 |
| Paradigm Shift | 3,400,000 | 761,719 | 0.650 |
| Grand Slam | 2,470,000 | 701,296 | 0.547 |
| Train Station | 16,700,000 | 8,755,605 | 0.280 |

towards the lower portion of the graph where experimental results are small. We leave the explanation of this for the discussion.

### Extending the Model for Three or More Words

To extend the model to three word combinations we go back to Equation (8) for $p(i)$ and plug in the same value for $I$ and our measured values for $\beta$ and $\alpha$:

$$p(i) = \left(\frac{1-\alpha\beta}{I^{1-\alpha\beta}-1}\right)\frac{1}{i^{\alpha\beta}} = \left(\frac{.48}{I^{.48}-1}\right)\frac{1}{i^{.52}}.$$

We then use the same linear approximation we did in the "Derivation of the Model" section:

$$p(A,B,C) = Ap(i)\cdot Bp(i)\cdot Cp(i) = ABC\cdot p^3(i) \approx ABC\cdot\left(\frac{.48}{I^{.48}-1}\right)^3\left(\frac{1}{i^{.52}}\right)^3$$

To find the expected number of results for the whole index, we take a sum and approximate it with an integral to find $R$,

$$\begin{aligned} R = \sum_1^I p(A,B,C) &\approx ABC\left(\frac{.48}{I^{.48}-1}\right)^3 \int_1^I \frac{di}{i^{1.56}} \\ &= ABC\left(\frac{.48}{I^{.48}-1}\right)^3\left(\frac{1}{.56I^{.56}} - \frac{1}{.56\cdot 1^{.56}}\right) = ABC1.072\cdot 10^{-15} \end{aligned}$$

$$\begin{aligned} R = \sum_1^I p(A,B,C) &\approx ABC\left(\frac{.48}{I^{.48}-1}\right)^3 \int_1^I \frac{di}{i^{1.56}} \\ &= -ABC\left(\frac{.48}{I^{.48}-1}\right)^3\left(\frac{1}{.56I^{.56}} - \frac{1}{.56\cdot 1^{.56}}\right) = \frac{ABC}{31,700,000^2} = \frac{1}{\left(I_{eff3}\right)^2}. \end{aligned}$$

Or:

$$I_{eff3} = 30,370,000.$$

We tested 89 word triplets and plot the results in Figure 6 at the same horizontal scale as Figure 5. The solid line is the theoretical expectation of $ABC$ given $R$, based on the calculated value of $I_{eff\ 3} = 30{,}370{,}000$. $(x,y) = (ABC,R)$. This behaviour suggests that our initial linear

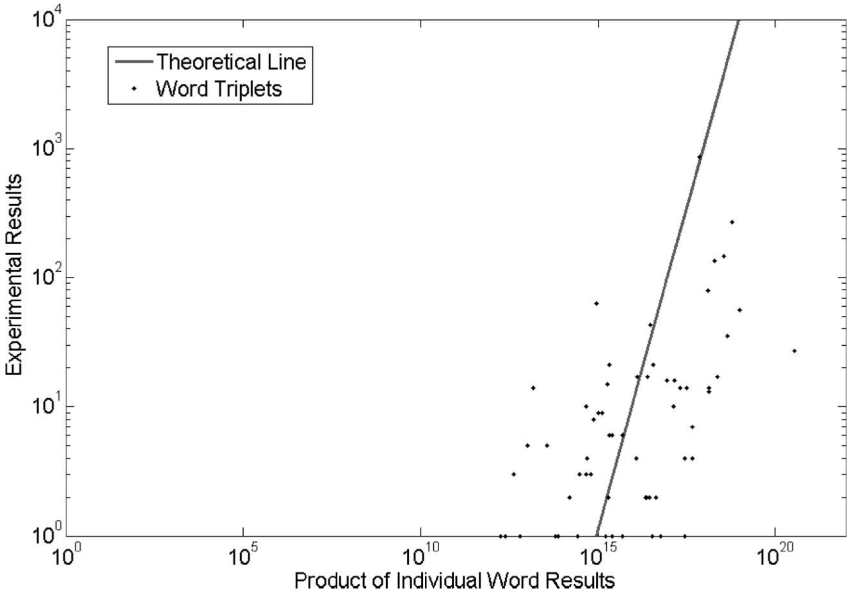


Fig. 6. This figure plots the results for all three words together versus the product of the three individual results.

approximations and general methods yield reasonable results up to searches for groups of three words.

The solid line is the theoretical expectation of *ABC* based on the calculated value of $I_{eff\,3} = 30{,}370{,}000$.

## The Dynamics of $I$ and $I_{eff}$

Since both $\beta$ and $\alpha$ are theoretically independent of the index size, we expect them to remain constant even as the index grows larger. We can solve for $I_{eff\,2}$ in terms of $I$ and vice versa by Equation (12).

$$\frac{(\alpha\beta - 1)^2 \left(I^{2\alpha\beta} - I\right)}{(2\alpha\beta - 1)(I^{\alpha\beta} - I)^2} = \frac{1}{I_{eff2}}$$

At the time of the first experiment (August 2005) Google's homepage displayed that it indexed around 8 billion pages. Since Yahoo's claim to have surpassed their index, Google has taken the number off to stress the importance of their page-rank sorting method rather than the sheer size of an index. At the time of the second experiment (April 2006) an approximate size of the total pages indexed could be obtained by a wildcard search of the two characters "**", which returns this number. At the time of the third experiment (June 2006) this trick no longer worked.

We ran an experiment in April of 2006 with the same words that were used 8 months earlier in August 2005 (described in the section "Application of the Model to Determining Relatedness of Words"). Google's index had approximately tripled during the time between experiments, growing from about 8 billion to about 25 billion based on the wildcard search terms "**". To make the plots simpler, we considered a range of values for the results of a search for two words (the $y$-values on the graph in powers of 10, i.e. from $10^n$ to $10^{n+1}$) and computed the geometric mean of the $x$-values (the product of $AB$ for each point). The new effective index size was calculated with Equation (12) (the value of $I$ was given from the Google website). In Figure 7, we plot the points along

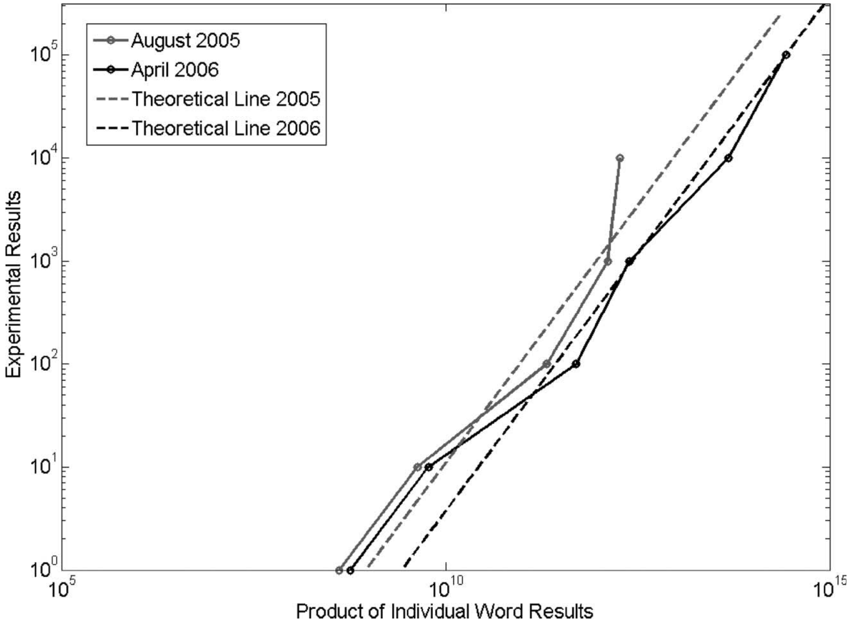


Fig. 7. Using the data from the section "Application of the Model to Determining Relatedness of Words", we considered a range of values for the results of a search for two words (the $y$-values on the graph are in powers of 10, i.e. from $10^n$ to 10 $^{n+1}$) and computed the geometric mean of the $x$-values (the product of $AB$ for each point). A second set of results is also plotted in the same manner and for the same set of words but 8 months later.

with the theoretical line they should lie on based on Equation (14) for August 2005 and April 2006.

## Comparing Yahoo and Google

In August of 2005 Yahoo claimed its index had far surpassed Google's. Previous comparative studies have looked at the overlap of pages indexed by search engines to determine which has the larger index (Bharat & Broder, 1998). Our method determines the size of each index directly and independently, thereby avoiding the bias of choosing a small set of data to analyze. If we assume that $\beta$ and $\alpha$ are the same for both engines, we can put the Yahoo index to the test. Figure 8 plots the results in the same manner as in Figure 7 except that the region in $y$ that is grouped is no longer from $10^n$ to $10^{n+1}$ but from some value $q$ to $q \cdot 10^{0.3}$ times the region below it. By choosing an effective index value which fits both curves reasonably, we show that both engines have approximately the

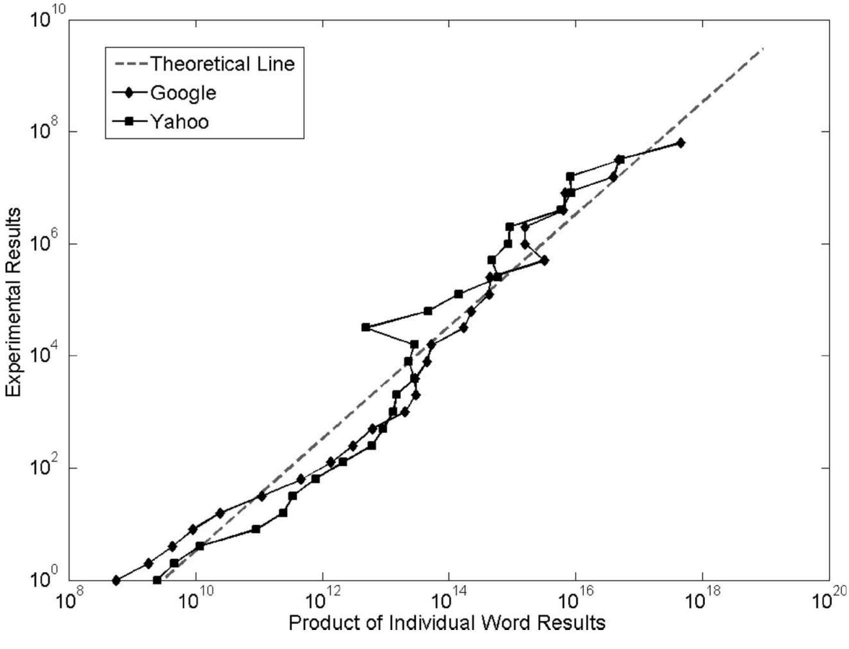


Fig. 8. In a similar manner to Figure 7, the geometric mean of word pair results is plotted versus the geometric mean of the product of individual results for sets of point points separated by power of $10^{0.3}$. Results are for the same word pairs searched on both engines at nearly the same time (in June 2006).

same index size. The actual value of $I$ can be found by plugging this effective index size into Equation (11). The numerical solution puts $I$ for both search engines at around 28,000,000,000 pages as of June 2006.

One difficulty in comparing search engines in this matter arises from changes in the Zipf law parameter $\alpha$, given that $\beta$ will be expected to stay the same as it is a function of the English language alone. Normally search engines only index pages up to a certain number of kilobytes (Bondar, 2006). This would put a cap on the page length and ruin the Zipf distribution making it look more uniform, the effective value of $\alpha$ would be smaller, which would then make $I_{eff\,2}$ look more like $I$. This would push the data points in Figures 5 to 8 to the right. Since Yahoo and Google may have different values for this cap on page size, this comparison may not be entirely fair.

## DISCUSSION

While these results appear promising we are left with a few pressing questions. Can we quantify the wide spread of results around the peak probability, and can we explain the distinct asymmetry of this spread toward lower numbers of results (see Figure 5)?

### Validation of Approximation for Exact Probabilities

We begin with Equation (2) which will return the exact probability density function (PDF) for the number of results when words with individual results $A$ and $B$ are spread on $I$ equal pages. Next we plug in $I_{eff\,2}$ for $I$, and examine the width of the peak to determine if our error is reasonable.

We verify the validity of this step with the computational model as shown in Figure 9.

Simulations were performed for the particular case of $A = B = 40$ and $I = 300$ with $\alpha\beta = 0.52$. The number of times both words appeared on the same page was recorded. The solid line gives the discrete PDF found based on these simulations. The dashed line is the approximate PDF with $I_{eff\,2}$ plugged into Equation (2) for $I$. For the same reasons as explained earlier, this approximation is only valid when $A, B \ll I$. We see from plots like these that the width of the peak in the simulation is similar to the width derived based on the model. The main difference between the two curves is that the theoretical curve is shifted to the right of the simulation

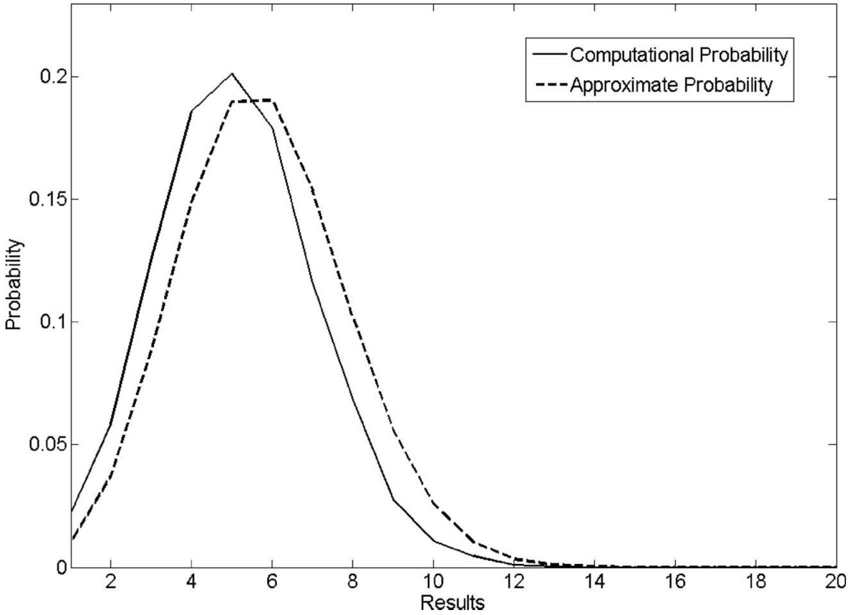


Fig. 9. The discrete probability density function (PDF) determined with the computational model is plotted as a solid line next to the approximate PDF using the effective index from Equation 11 values for $I$ in Equation 2 for the exact probabilities in a uniform distribution. The values used were: $A = B = 40$, $I = 300$, $\alpha\beta = 0.52$.

curve. This shift quickly becomes negligible as $A$ and $B$ become small fractions of $I$ and so this approximation is valid for our purposes.

## Expected Results and Corresponding Asymmetry

A quick study shows that this peak shown in Figure 9 is always very sharp for large $I$. This is not surprising because functions with factorials often have very sharp peaks (as in statistical physics). To demonstrate this with an example, we consider the word pair "Psychometric" and "Carpaccio". Summing probabilities (using Equation [2]) shows the number of results should be somewhere between 600 and 800 with 99.99% probability. The actual number of results returned was 12. Differences like this are not uncommon.

In Figure 10 we plot the data points collected for the same type of experiment for the same word pairs as in Figure 5. In the same plot we present a contour plot for the theoretical probabilities computed from Equation (2) with $I_{eff2}$. For the sake of simplicity we use $A = B$ in

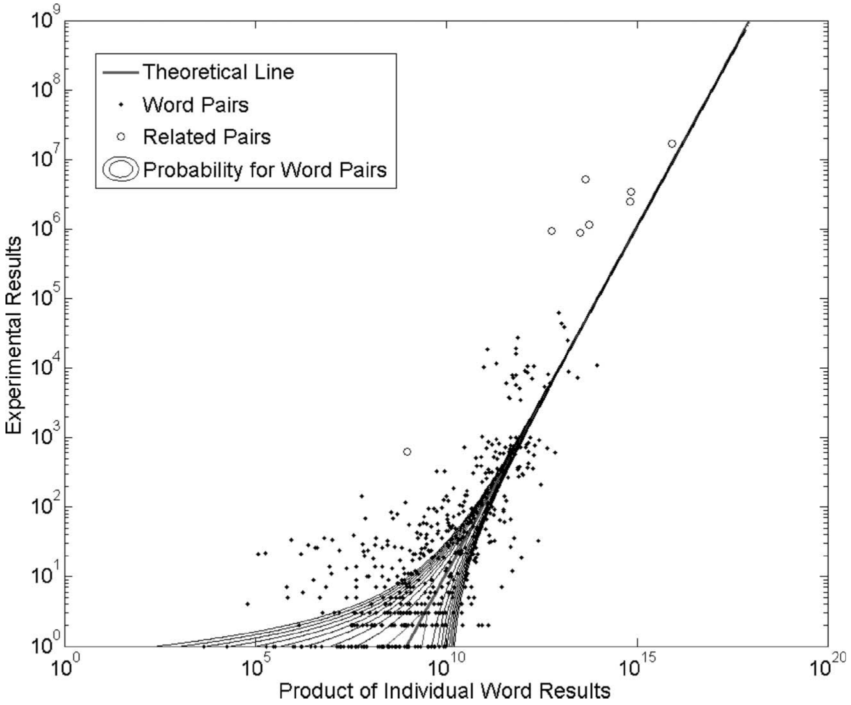


Fig. 10. The discrete probability density function determined with the computational model is shown as a contour plot below the experimental data points. The regions outside the contours have a probability lower than $10^{-15}$. Associated word pairs are marked as circles.

Equation (2) which will give us the widest possible distribution in $R$ for a given value of the product $AB$. It is clear from the figure that many of the points lay in regions with very low probabilities. The regions outside the contour lines have a probability lower than $10^{-15}$.

We will now discuss the symmetry of the points and return to the description of these errors later.

It is clear in Figures 5 and 10 that the points are not spaced symmetrically around the (solid) theoretical line. Word pairs with a low product of individual word results appear to have much higher results together than expected (many points lie above the theoretical line). The reason for the asymmetry is two-fold. First the asymmetric data springs directly from an asymmetric probability distribution as the contours in Figure 10 indicate. Also, since Googlewhacks are often rare words, the majority of the words from the experiment had low results. A full quarter of the word pairs had expected and exact results of essentially zero. While this effect is not enough alone to explain the appearance of points in sections with values corresponding to absurdly low probabilities, the tendency can be exaggerated by the error-increasing effects we will discuss next.

**Strength of Associativity**

We have assumed that the probability of a word appearing on a page is largely independent of the word in question. It is entirely possible that two words are so disassociated from each other that the presence of "Zugzwangs" on a page (for example) reduces the probability that "Plankton" will be on it as well. One word increasing the probability of another occurs quite frequently, as we have seen with the tests of associated words. If this is the largest reason for error, then in fact it is not error at all, simply an accurate test of "strength of associativity".

**Some Little-Known Google Attributes**

In order to carefully test our model, we have compared it to responses given by Google, results we have assumed to be approximately correct. Google always displays "approximate results" for large returns, the variation from expected results in our experiments is often far above this precision level.

It is important to note that there is a difference between what we call a result, and a result which Google returns. Google's algorithm is more suited to practical searching than linguistic research. A search for the two words "miserable" and "failure" together became famous for giving the whitehouse.gov biography of George Bush as the first hit while neither of those words appears on that webpage. It appeared as a hit because a large number of sites linked to the biography with "miserable failure" in the anchor text.

We have considered two methods to test the size of these effects and to show that the deviation from predictions is sufficient to explain the differences we have found between our experimental and theoretical results. The most direct way is to switch the order of the search terms. For example, in February 2008, searching for "Goddamned Toolboxes" returned 333,000 results but "Toolboxes Goddamned" returned only 37,500. Since the word order does not affect the total number of pages they actually appear together on the number of results should have been the same. Further testing revealed that for various word pairs these two values are off by a factor of one and a half times, on average.

Another method makes use of Google's advanced search options. The set of pages with "miserable", but without "failure" can be found by searching for "miserable – failure". We will denote the number of pages in this set as {a – b}, the number of pages with both the words as {a b} and the number of pages with just the first word as {a}. In this case the following formula should hold:

$$\{a - b\} + \{b - a\} + \{ab\} = \{a\} + \{b\} - \{ab\}$$

or:

$$(1/2)(\{a\} + \{b\} - \{a - b\} - \{b - a\}) = \{ab\}$$

Comparing the result from this test with the actual result from Google for {a b} shows that most values on either side of the equal sign are off by around a factor of $10^{2.6}$ with some off by as much as $10^{5.6}$. Although this short test does not directly correlate to the problem at hand, the order of magnitude of the change implies that there may be similarly-sized changes for the results used in our experiment. This difference is more than enough to explain the wide distribution we found in our experiments.

## CONCLUSION

We have developed a model to predict the number of results that should be returned by a pair of words entered into the Google search engine based on the Zipf law for the distribution of website text sizes, and Heaps' law for the vocabulary size in those documents. We use data from 351 word pairs that return exactly one result to easily measure the Heaps' law parameter $\beta = 0.52$ for a library of over 8 billion pages.

We demonstrate the validity of our methods by obtaining reasonable results after extending the model to word triplets. We also confirm the model over a period of 8 months during which the Google index size tripled. We then compare the index sizes of the two competing search giants Yahoo and Google and find that they both have about the same index, a size that we estimate to be around 28 billion pages. This information is no longer public but we believe we are the first to have a method of measuring this indirectly.

We mention that tests we have conducted show that internet searches for random numbers return approximately log-normally distributed results and sorting by first digits yields results that follow Benford's law (1938). This is an area of study of internet behaviour that is worth further investigation.

Potentially the most useful part of our paper is our simple method for automatically determining the "strength of associativity" of various word pairs. This test requires a tiny amount of computational power and can thus be used for a huge list of word pairs in a short time or for single word pairs with immediate results. The time to test each pair is essentially limited only by the time required for three Google searches (fractions of a second). A more accurate model will, of course, result in a more accurate value of "strength of associativity".

We identified that the main deviations from the model derive from idiosyncrasies in how Google returns results. Future models must take into account reasons for this, such as results returned because of linking anchor text rather than the text on the given page. It may be possible to extrapolate the real number of results from a clever combination of searches using the advanced search options. Such a model would provide more accurate measures for all the values determined in this paper, including the Heaps' law parameter, index size, and "strength of associativity".

As the Internet continues to grow, it will be interesting to see whether the Zipf's and Heap's law parameters determined in this paper will change. A more detailed comparative study can be easily performed using Google's advanced search options to determine the parameters for specific languages or countries. We are fortunate that the advent of the internet has made studies like these efficient, where they would have been impossibly difficult to even consider as little as 10 years ago.

## REFERENCES

Adamic, L. A., & Huberman, B. A. (2002). Zipf's law and the internet. *Glottometrics*, *3*, 143–150.

Altmann, G. (1988). *Wiederholungen in Texten*. Bochum: Studienverlag Brockmeyer.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval Web Usage Mining in Search Engines*. New York: Addison-Wesley.

Benford, F. (1938). The law of anomalous numbers. *Proceedings of the American Philosophical Society*, *78*(4), 551–572.

Bharat, K., & Broder, A. (1998). A technique for measuring the relative size and overlap of public web search engines. *Proceedings of the 7th International World Wide Web Conference,* Brisbane, Australia (WWW7), 379–388.

Bondar, S. (2006). Search engine index limits: Where do the bots stop? Retrieved September, 2007, from http://www.sitepoint.com/article/indexing-limits-where-bots-stop

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, *30*(1–7), 107–117.

Bukiet, E. (2005). Private communication. Ithaca, New York. Unpublished.

Downey, A. B. (2001). The structural cause of file size distributions. In *Joint International Conference on Measurement and Modeling of Computer Systems. Proceedings of the 2001 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (pp. 328–329). Cambridge, MA: ACM.

Faloutsos, M., Faloutsos, P., & Faloutsos, C. (1999). On power-law relationships of the Internet topology. *ACM SIGCOMM Computer Communication Review*, *29*(4), 251–262.

Feller, W. (1968). Stirling's formula. §2.9. In *An Introduction to Probability Theory and its Applications*, Volume 1, 3rd edition (pp. 50–53). New York: Wiley.

French, J. C. (2002). Modeling web data. *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital Libraries,* Portland, Oregon, 320–321.

Heaps, H. S. (1978). *Information Retrieval: Computational and Theoretical Aspects*. New York: Academic Press.

Weisstein, E. W. (1999). Hypergeometric distribution. From MathWord – A Wolfram web resource. Retrieved January, 2008, from http://mathworld.wolfram.com/HypergeometricDistribution.html

Ziegler, A., & Altmann, G. (2002). *Denotative Textanalyse*. Berlin: Edition Präsens.

Zipf, G. (1932). *Selective Studies and the Principle of Relative Frequency in Language*. Cambridge, MA: Harvard University Press.