Estimating the ImpressionRank of Web Pages

FAN Kai
fankai@net.pku.edu.cn

Computer Networks and Distritued Systems Laboratory
Peking University

May 15, 2009

ImpressionRank

- A measure of **exposure** of a web page/site in a search engine
- Number of times users viewed the page while browsing search results
- Page **p** has an impression on a query(keyword) **q** :
  - the search engine return **p** as a result for **q**
  - the user looked at the result(click is not necessary)
  - **the top n ranking results**

Motivation

- Popularity rating of pages and sites
  - assume visibility is corelated with traffic
  - Nielson, comScore and Alexa
- Site analytics
  - impressions vs. clicks
- Market research
  - different sites
  - different queries
  - different search engines
- Search engine evaluation
  - impressions of spams, hate sites, porn and virus infected pages

Contribution

- First **external** algorithm for popular keyword extraction
  - the search engine
  - the query suggestion service
  - the web
- Moderst Resource
  - search engine requests(hundreds)
  - suggestion service requests(tens of thousands)
  - fetch web pages(hundreds)

Challenges

- find queries for which the search engine would return a certain page
- find how many impressions these queries generate
- limits on the rate of requests posed by search engine

ImpressionRank Calculation

- Notions
  - **impression(p,w)** impression contribution of keyword **w** to page **p**
  - **incident(p,w)** whether the search engine return **p** for **w**
  - **freq(w)** number of **w** in the query log
  - **N(p)** neighbors of page **p** , or all keywords incident to **p**
- Formula
  - $impression(p, w) = incident(p, w) \cdot freq(w)$
  - $irank(p) = \sum_w impression(p, w) = \sum_{w \in neighbor(p)} freq(w)$
  - $irank(website) = \sum_{p \in website} irank(p)$

Popular Keyword Extraction

- Power law distribution of query frequencies
  - 73% impressions of www.cnn.com come from "cnn", "election results", "news"
  - assume power law holds for a specific document
- Varient of classical keyword extraction problem
  - find impressions on top queries
- Algorithm
  - use the frequencies of the top k keywords to infer the exponent of power law
  - sum up the total frequency to estimate the ImpressionRank

Remaining Problems

- Keyword Frequency Estimation

  Input A keyword **w**

  Output The frequency of **w** in search engine log

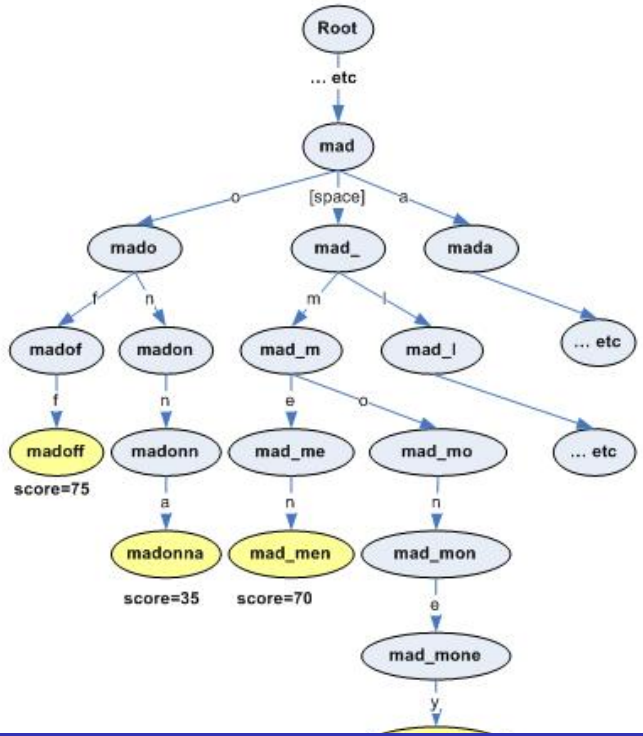
- Popular Keyword Extraction

  Input A document **p** and an integer **k**

  Output The **k** keywords on which **p** has the most impressions

Suggestion Services

- Given a string **s** , suggesting service returns the most popular queries which starts with **s** .
- *VLDB 2008, Mining Search Engine Query Logs via Suggestion Sampling*



Frequency vs. Volume

- Notions
  - freq(q) frequency of a query **q**
    number of instances **q** *in the query log*
  - vol(s) *volume of a string* **s**
    number of distinct queries in the log of whom **s** is a prefix
  - pre(q) shortest exposing prefix of a query **q**
    shortest prefix **s** *of* **q** *for which the suggestion service return* **q** *as one of the suggestions for* **s**
- Correlation between **freq(q)** and **vol(pre(q))**
  - both have power law distributions
  - "order-corelated"
  - measure **freq(q)** using **vol(pre(q))**
- How to find **pre(q)**?

Naive Volume Calculator

- Assume suggestion service return M results at most each time.
- If we send a string **s** to suggestion service and the service return $k(< M)$ results, we could infer $vol(s) = k$.
- If suggestion service return$k(\geq M)$ results, recursively computes the volumes of the children of **s** and sum them up.

Real Volume Estimator

- Naive estimator

$$VolEst_{naive}(s) = \begin{cases} |results(s)|, & \text{if } |results(s) < M \\ a, & \text{if } k \geq M \end{cases}$$

- Sample-based estimator
  given $T'$is a random sample of T

$$VolEst_{sample}(s) = \begin{cases} vol(s, T') \cdot \frac{|T|}{|T'|}, & \text{if } |vol(s, T') \geq b \\ 0, & otherwise \end{cases}$$

- Score-based estimator

Theorem

$$VolEst(s) = VolEst_{naive}(s) + VolEst_{sample}(s) + VolEst_{score}(s)$$

Overview

- Impossible to test all keywords.
- Apply *best-first search*
  - *track down the most promising candidates*
  - *evaluate them*
  - *report the top keywords found*
- Cache everything.
- Set requests budget.

Notions

- **candidate heap** : frontier of the search space
- **keyword heap** : highest frequent keywords incident to document
- **seed text** : all related text to a web page
- **term pool** : all terms from **seed text**
- **candidate keywords** : all ordered finite-length sequences of terms from **term pool**

search space : a TRIE whose alphabet is all the terms

Main Flow

1. crawl seed text, add all terms to **term pool**
2. **score** all terms and insert them to **candidate heap**
3. loop while budget not reached
   1. pop **w** from **candidate heap** and send to suggestion service
   2. $S = \mathbf{w} \cup result(w)$
   3. for all $u \in S$, if u is **incident** to page **p**
      1. **estimate** freq(u) and add to **keyword heap**
      2. expand seed text with u and all pages incident to u
      3. regenerate all terms, **recore**, and refresh **candidate heap**
4. expand **w**

Test Incidence

- Send keyword **w** to search engine and check whether page **p** is one of the top k results.
- Send query "inurl:url(p) w" to the search engine, if no results returned, then neither **w** nor any of its descendants are incident to **p** .

Whether To Expand A Candidate

- diversity: if **w** or any suggestions for **w** already in **keyword heap** , no need to expand
- if none of the descendants of **w** has positive frequency, prune
- if none of the descendants of **w** is incident to **p** , prune
- if estimated frequency of the top descendant can not make top k in **keyword heap** , prune
- otherwise, expand **w**

Candidate Scoring

- efficiency is important
- $score(w) = score_{freq}(w)^a + score_{tf}(w)^b + score_{idf}(w)^c$
- estimate frequency score for w and all its descendants
  - under budget
  - estimate frequency for keywords r which has most descendants
    - if vol(r) is 0 or 1, then all its descendants has no frequency
    - if r is choosed, then its ancestors is also choosed
  - for keyword u which is not selected
    - if r=pre(u) is selected, estimate its score by vol(pre(u))
    - if r=pre(u) is not selected, find its lower-most ancestor r which has been selected, F is the sum of the frequencies of r's descendants not in suggestions, $score_{freq}(u) = \frac{F}{vol(r)-N}$

Query Popularity

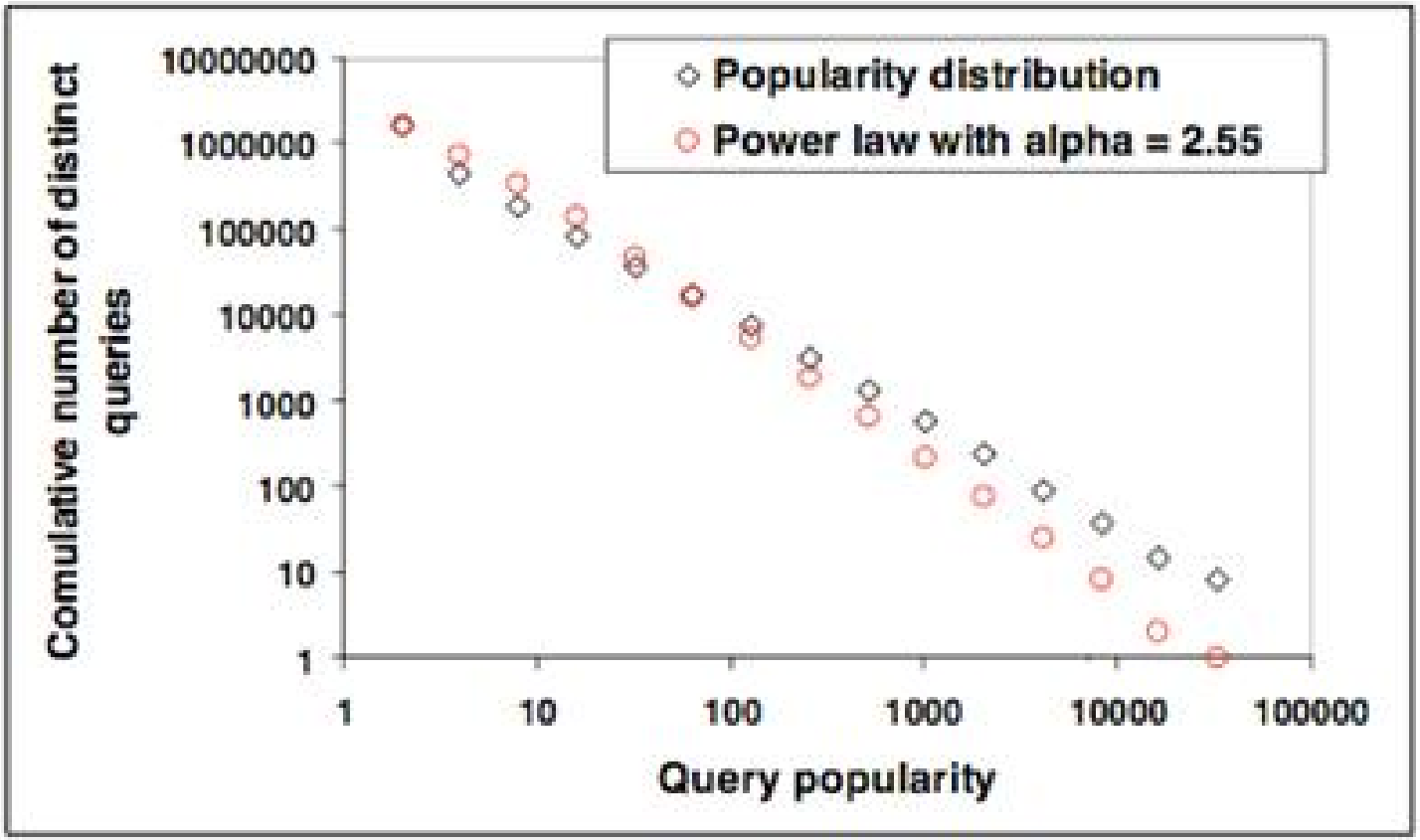


Figure 2: Cumulative number of distinct queries by popularity in the AOL data set (log-log scale). The graph indicates an almost power law with exponent $\alpha = 2.55$.

Recall

α, β, γ	recall_F, **Google**	recall_F, **Yahoo!**	recall_U, **Google**	recall_U, **Yahoo!**
$0.2, 1, 0.6$	0.93 ± 0.08	0.84 ± 0.14	0.62 ± 0.06	0.37 ± 0.05
$0, 1, 0.6$	0.91 ± 0.09	0.80 ± 0.16	0.52 ± 0.06	0.27 ± 0.04
$0.2, 0, 0.6$	0.02 ± 0.01	0.07 ± 0.08	0.24 ± 0.05	0.17 ± 0.04
$0.2, 1, 0$	0.92 ± 0.08	0.82 ± 0.14	0.50 ± 0.06	0.17 ± 0.04

Table 1: Estimated recall values (together with measured standard deviations) for the popular keyword extraction algorithm.

Recall

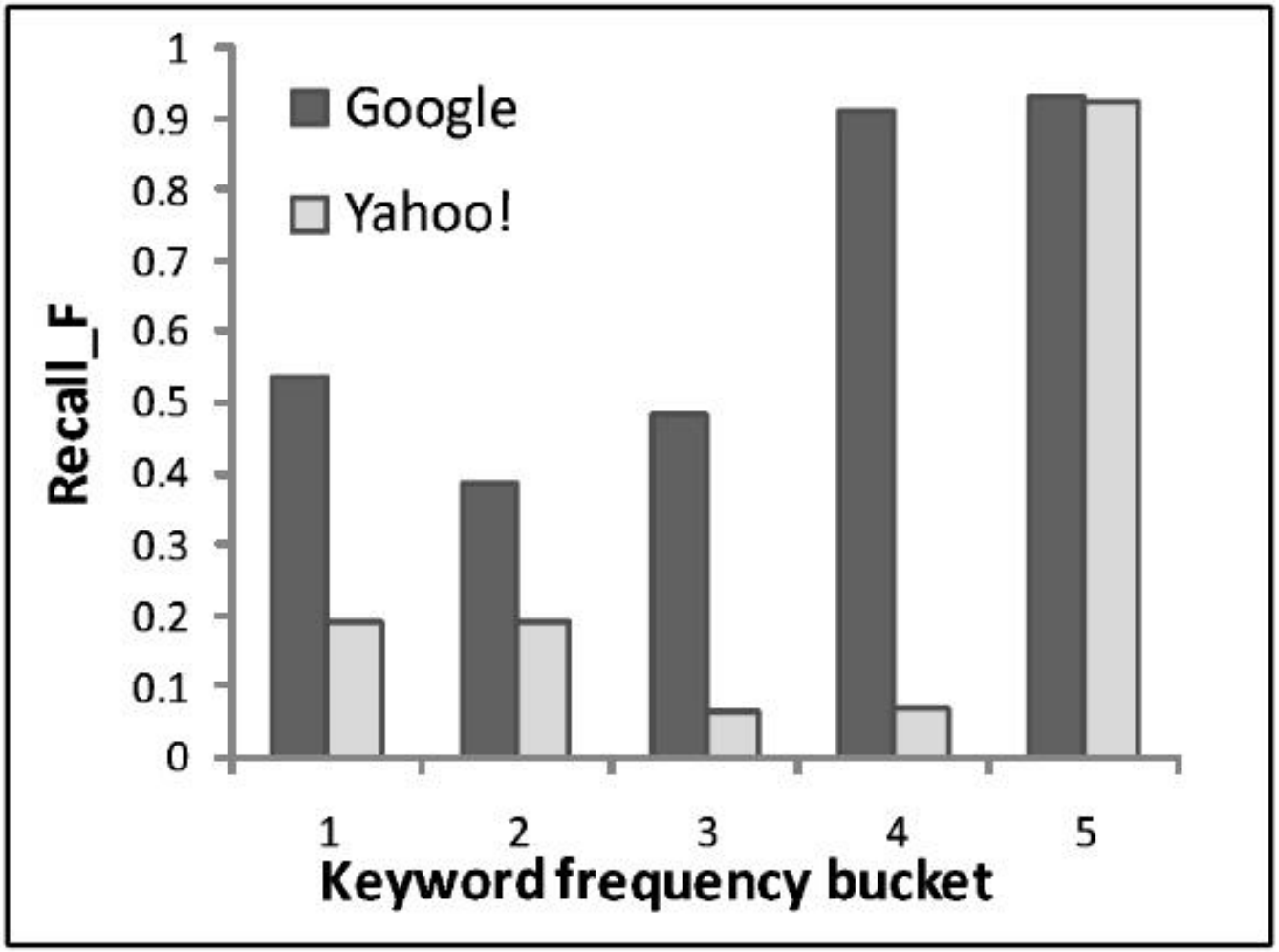


Figure 2: Recall as a function of keyword frequency.

Progress

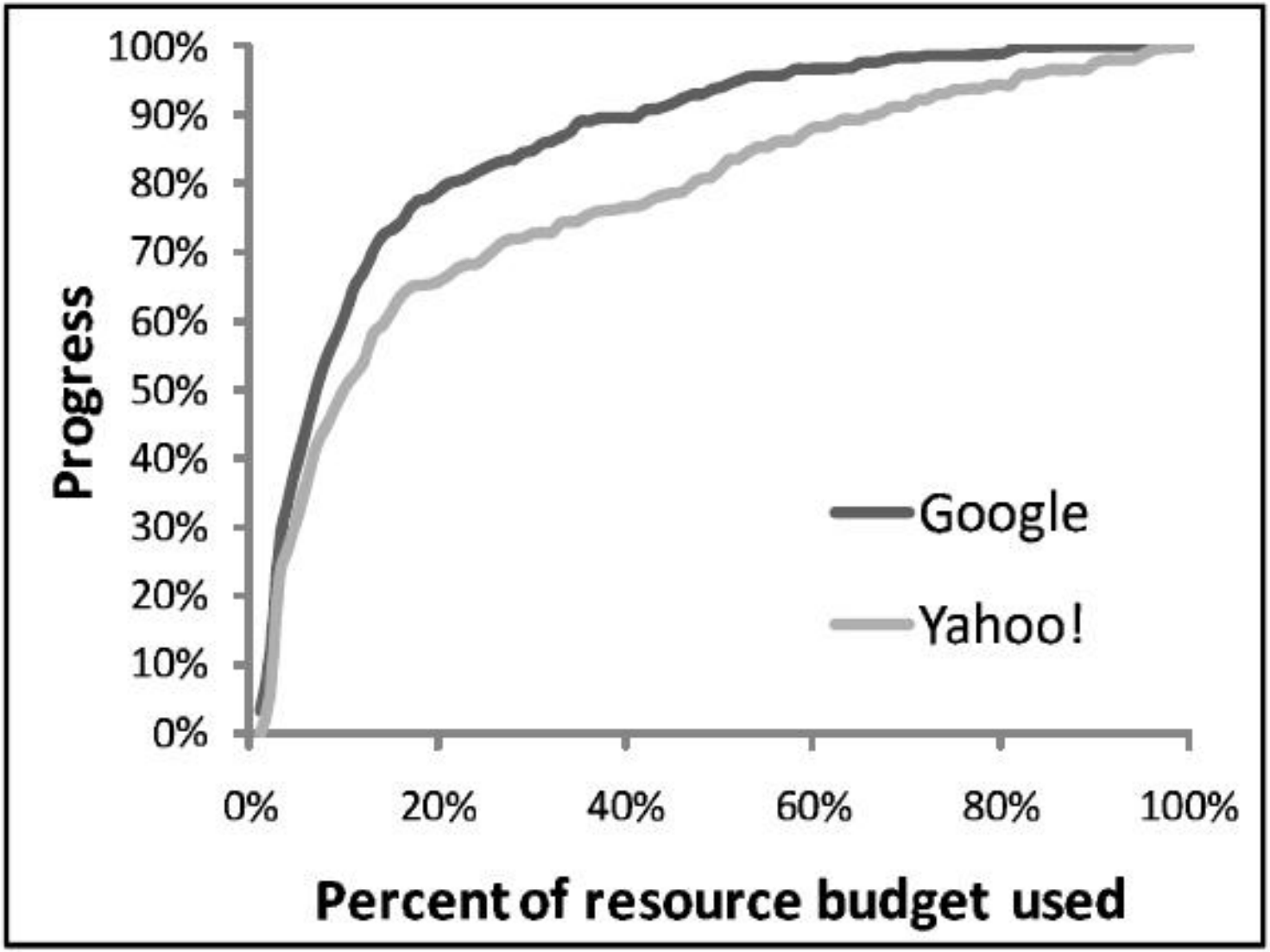


Figure 3: Keyword extraction progress.

ImpressionRank Estimates For News Sites

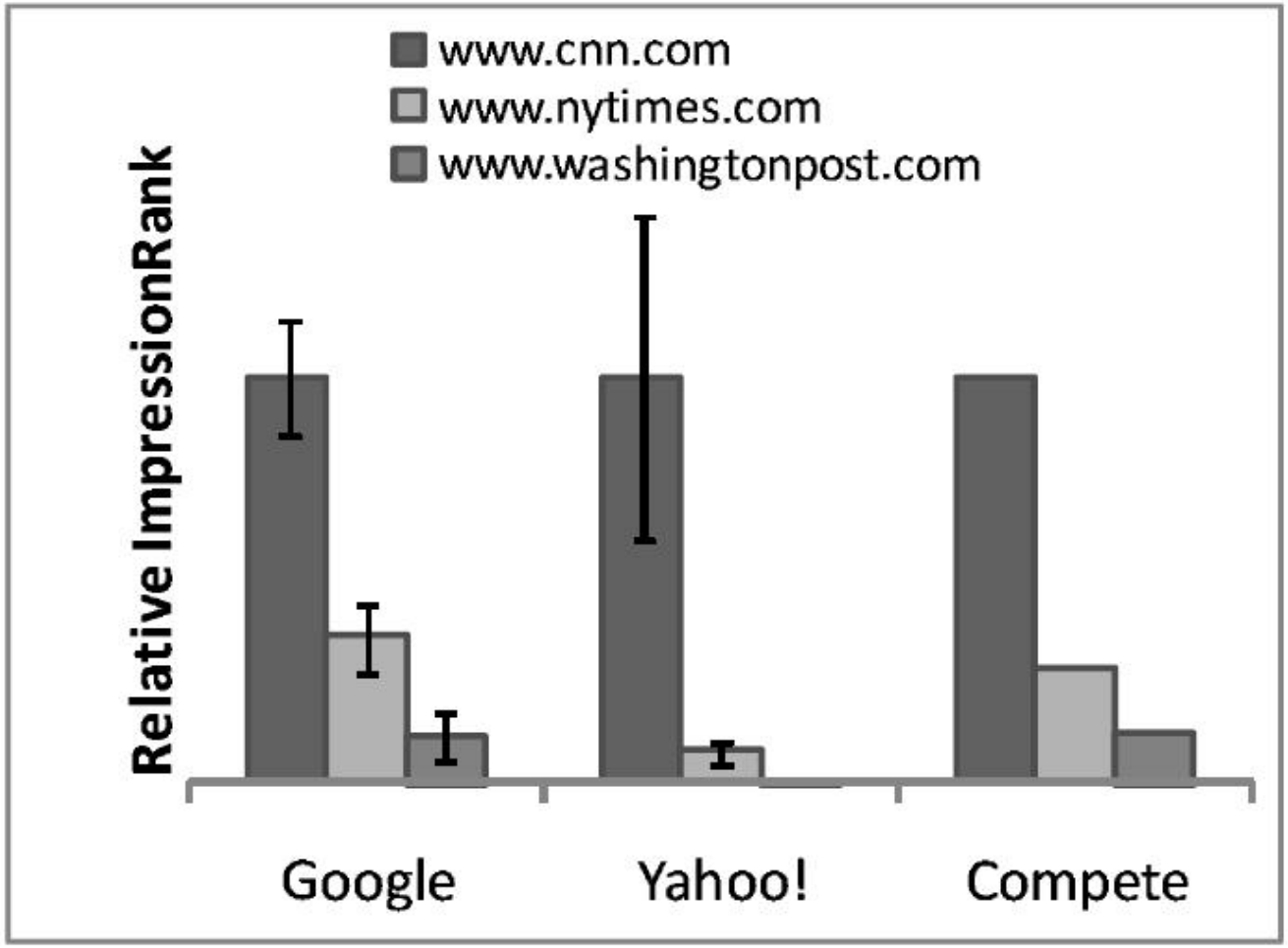


Figure 4: ImpressionRank estimates for news sites.

ImpressionRank Estimates For Travel Sites

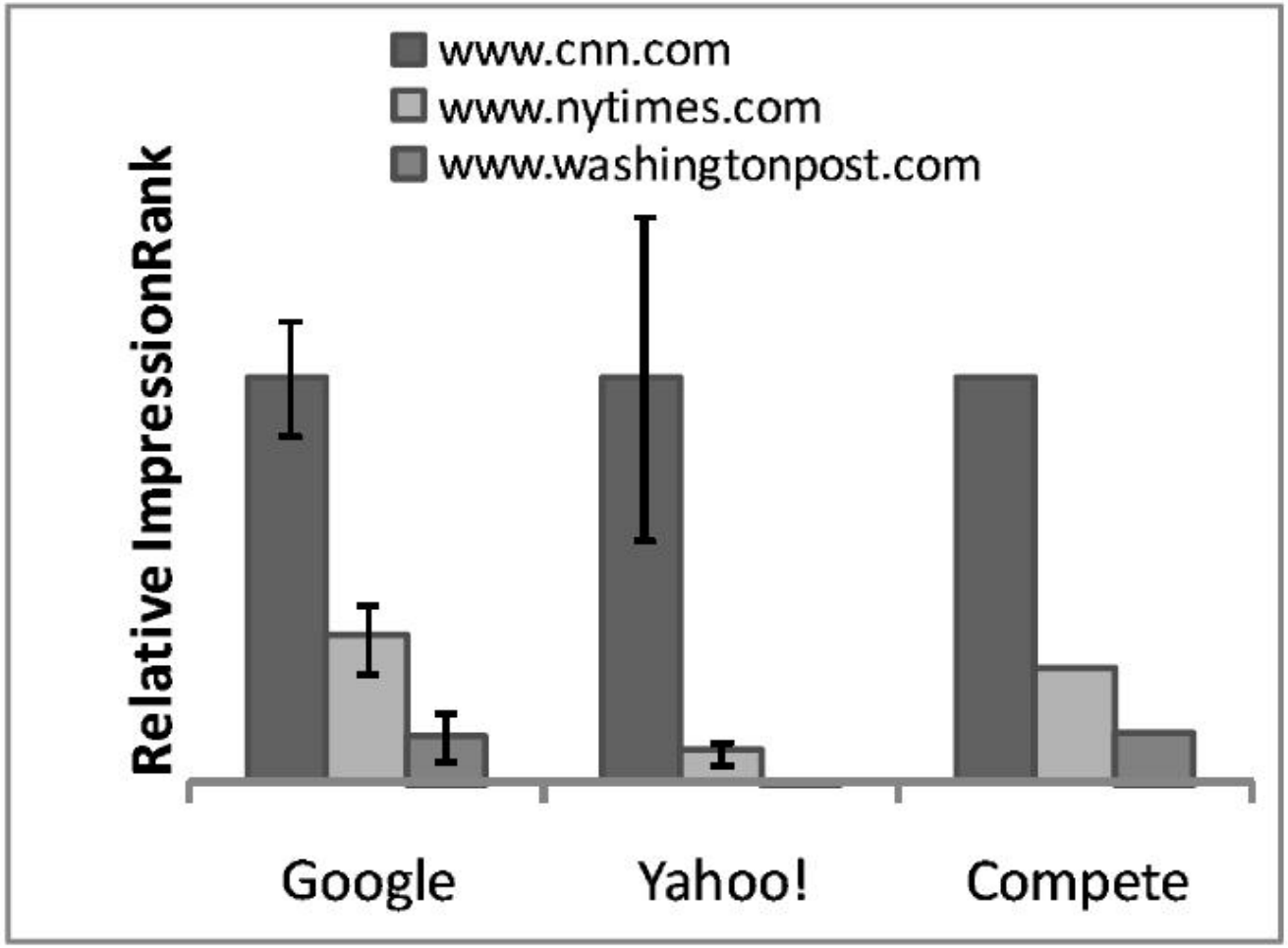


Figure 4: ImpressionRank estimates for news sites.

Pupular Keywords

Google	Yahoo!
www.cnn.com	
cnn, election results, news, obama, video, polls, health	weather, cnn, news of the world, obama, cnn news, presidential election
www.nytimes.com	
new york times, fashion, obama girl, crossword puzzles	new york times, ny times, crossword, the new york times, new times
www.washingtonpost.com	
obama tax plan, washington post, washington, post, the post	washington post, comics, newspaper, iraq news
en.wikipedia.org/wiki/PageRank	
page rank, ranking, google ranking, google page rank, pagerank	pagerank, google algorithm, google rank, page rank
www.expedia.com	
expedia, travel, travel agents, ski holidays, cruises, cruise deals	expedia, hotels, cheap hotels, travel, vacation packages
www.orbitz.com	
orbitz, car rental, cheap hotels, flights, airline tickets,	hotels, cheap tickets, cheap flights, orbitz, travel, flights
www.travelocity.com	
travelocity, travel, flights, flight tickets, travelocity flights,	hotels, cheap flights, travelocity, travel, flights, airline tickets
www.cs.cornell.edu/home/kleinber	
scientific american articles, kleinberg, kleinberg tardos, small world phenomenon	kleinberg, small world phenomenon, jon kleinberg, world phenomenon, robin ec08 parts
www.microsoft.com/BillGates	
bill gates, bill gates home, william gates, william gates iii bill gates speeches,	gates, william gates, william gates iii, pictures of bill gates house
infolab.stanford.edu/~sergey	
sergey brin, favorite books, brin sergey, stanford sergey brin, data mining search engines	sergey brin, brin sergey, stanford home page, cs stanford, brin page, page brin

Think Boldly!

Thanks