Relevance Feedback in Web Search

Sergei Vassilvitskii (Stanford University)
Eric Brill (Microsoft Research)
Introduction

• Web search is a non-interactive system.
• Exceptions are spell checking and query suggestions
• By design search engines are stateless

• But many searches become interactive:
  • query, get results back, reformulate query...
  • Can use interaction to retrieve user intent
Relevance Feedback

Special Inspector General for Iraq Reconstruction: SIGIR Homepage
Welcome to the Office of the Special Inspector General for Iraq Reconstruction (SIGIR), a temporary federal agency serving the American public as a watchdog for fraud ...

ACM SIGIR Special Interest Group on Information Retrieval Home Page
Welcome to the ACM SIGIR Web site. ACM SIGIR addresses issues ranging from theory to user demands in the application of computers to the acquisition, organization, storage, retrieval, and ...
www.sigir.org

SIGIR 2004
The major international forum for the presentation of new research results and demonstration of new ... SIGIR is the major international forum for the presentation of new research results and the ...
www.sigir.org/sigir2004

SIGIR 2006—Seattle
29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval, Seattle 2006 ...
www.sigir2006.org
Using This Information

- Classical methods: e.g. Rocchio’s term reweighing (TFiDF) + cosine similarity scores.

- There is more information here: what can the structure of the web tell us?
Hypothesis

- For a given query:
  - Relevant pages tend to point to other relevant pages.
    - Similar to Pagerank.
Hypothesis

• For a given query:
  • Relevant pages tend to point to other relevant pages.
    ➡ Similar to Pagerank.
  • Irrelevant pages tend to be pointed to by other irrelevant pages.
    ➡ “Reverse Pagerank”
    ➡ Those who point to web spam are likely to be spammers.
Dataset

- Dataset
- 9500 queries
- For each query 5 - 30 result URLs
- each URL rated on a scale of 1 (poor) to 5 (perfect)
- Total 150,000 (query, url, rating) triples

- Will use this data to simulate relevance feedback
- Only reveal the ratings for some URLs
Hypothesis Validation

- Relevance distribution of all URLs in the dataset
Hypothesis Validation

- Relevance distribution of all URLs in the dataset
- Compared to the URLs that are targets of perfect results

![Bar chart showing baseline and perfect targets relevance distribution](image)
Towards an Algorithm

- url₁
- url₂
- url₃
- url₄
- url₅
- url₆
Towards an Algorithm

url1
url2
url3
url4
url5
url6

unrated result
good result
bad result
Towards an Algorithm

url1

url2

url3

url4

url5

url6

unrated result

good result

bad result
Towards an Algorithm
Towards an Algorithm

- $url_1$
- $url_2$
- $url_3$
- $url_4$
- $url_5$
- $url_6$

- $url_2$
- $url_6$
- $url_1$
- $url_4$
- $url_3$
- $url_5$

- unrated result
- good result
- bad result
Percolating the Ratings

- Calculate the effect on $u$
- Begin with a probability distribution on relevance of $u$ (Baseline histogram)
- For all highly rated documents $v$
  - If there exists a short $v \rightarrow u$ path, update $u$.
- For all irrelevant documents $v$
  - If there exists a short $u \rightarrow v$ path, update $u$.
- Combine the static score together with the relevance information
Algorithm Parameters

• If there exists a “short” path...
  • Strength of signal decreases with length
  • Recall of the system increases with length
  • Computational considerations
  • Looked at paths of 4 hops or less
Algorithm parameters

- If there exists a “short” path...
  - Strength of signal decreases with length
  - Recall of the system increases with length
  - Computational considerations
  - Looked at paths of 4 hops or less
- ...update $u$.
- Maintain a probability distribution on the relevance of $u$. 
**Experimental Setup**

- For each query in the dataset split the URLs into:
  - Train: the relevance is revealed to the algorithm
  - Test: Only the static score is revealed

- Compare the ranking of the test URLs by their static score vs. static + RF scores.
**Evaluation Measure**

- Measure: NDCG (Normalized Discounted Cumulative Gain):

  \[ NDCG \propto \sum_i \frac{2^{rel(i)} - 1}{\log(1 + i)} \]

- Why NDCG?
  - sensitive to the position of highest rated page
  - Log-discounting of results
  - Normalized for different lengths lists
Result Summary

- NDCG change for three subsets of pages.
- Complete Dataset

**Roccio**: Demotes the best result
**Result Summary**

- NDCG change for three subsets of pages.
- Complete Dataset
- Only queries with NDCG < 100
**Result Summary**

- NDCG change for three subsets of pages.
- Complete Dataset
- Only queries with NDCG < 100
- Only queries with NDCG < 85

Increased performance for harder queries
Result Summary (2)

- Recall for the three datasets.
- Complete Dataset
- Only Queries with NDCG < 100
- Only Queries with NDCG < 85

![Bar Chart]

Legend:
- Alg
- Rocchio
Results Summary (3)

- Many more experiments:
  - How does the number of URLs rated affect the results?
  - Are some URLs better to rate than others?
  - Can we predict when recall will be low?
Future Work

- Hybrid Systems: Combining text based and link based RF approaches
- Learning feedback based on clickthrough data
- Large scale experimental evaluation of different RF approaches
Thank You

Any Questions?