Search Engines

- Provide a ranked list of documents.
- May provide relevance scores.
- May have performance information.
External Metasearch

Metasearch Engine

Search Engine A
Database A

Search Engine B
Database B

Search Engine C
Database C
Internal Metasearch

Search Engine

Metasearch core

Text Module

URL Module

Image Module

HTML Database

Image Database
Metasearch Engines

- Query multiple search engines.
- May or may not combine results.
Outline

✓ Introduce problem
  ■ Characterize problem
  ■ Survey techniques
  ■ Upper bounds for metasearch
Characterizing Metasearch

- Three axes:
  - common vs. disjoint database,
  - relevance scores vs. ranks,
  - training data vs. no training data.
Axis 1: DB Overlap

- High overlap
  - data fusion.
- Low overlap
  - collection fusion (distributed retrieval).
- Very different techniques for each...
- Today: data fusion.
Classes of Metasearch Problems

<table>
<thead>
<tr>
<th></th>
<th>no training data</th>
<th>training data</th>
</tr>
</thead>
<tbody>
<tr>
<td>ranks only</td>
<td>Borda, Condorcet, rCombMNZ</td>
<td>Bayes</td>
</tr>
<tr>
<td>relevance scores</td>
<td>CombMNZ</td>
<td>LC model</td>
</tr>
</tbody>
</table>
Outline

✓ Introduce problem
✓ Characterize problem
  ■ Survey techniques
  ■ Upper bounds for metasearch
## Classes of Metasearch Problems

<table>
<thead>
<tr>
<th></th>
<th>no training data</th>
<th>training data</th>
</tr>
</thead>
<tbody>
<tr>
<td>ranks only</td>
<td>Borda, Condorcet, rCombMNZ</td>
<td>Bayes</td>
</tr>
<tr>
<td>relevance scores</td>
<td>CombMNZ</td>
<td>LC model</td>
</tr>
</tbody>
</table>
CombSUM [Fox, Shaw, Lee, et al.]

- Normalize scores: [0,1].
- For each doc:
  - sum relevance scores given to it by each system (use 0 if unretrieved).
- Rank documents by score.
- Variants: MIN, MAX, MED, ANZ, MNZ
CombMNZ [Fox, Shaw, Lee, et al.]

- Normalize scores: $[0,1]$.
- For each doc:
  - sum relevance scores given to it by each system (use 0 if unretrieved), and
  - multiply by number of systems that retrieved it (MNZ).
- Rank documents by score.
How well do they perform?

- Need *performance metric*.
- Need *benchmark data*.
## Metric: Average Precision

<table>
<thead>
<tr>
<th>R</th>
<th>N</th>
<th>1/1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/3</td>
<td>N</td>
<td>2/3</td>
</tr>
<tr>
<td>3/5</td>
<td>N</td>
<td>3/5</td>
</tr>
<tr>
<td>4/8</td>
<td>N</td>
<td>4/8</td>
</tr>
</tbody>
</table>

$$0.6917$$
Benchmark Data: TREC

- Annual *Text Retrieval Conference*.
- Millions of documents (AP, NYT, etc.)
- 50 queries.
- Dozens of retrieval engines.
- Output lists available.
- Relevance judgments available.
## Data Sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Number systems</th>
<th>Number queries</th>
<th>Number of docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC3</td>
<td>40</td>
<td>50</td>
<td>1000</td>
</tr>
<tr>
<td>TREC5</td>
<td>61</td>
<td>50</td>
<td>1000</td>
</tr>
<tr>
<td>Vogt</td>
<td>10</td>
<td>10</td>
<td>1000</td>
</tr>
<tr>
<td>TREC9</td>
<td>105</td>
<td>50</td>
<td>1000</td>
</tr>
</tbody>
</table>
CombX on TREC5 Data

TREC 5: Combining the top i systems in order.
CombX on TREC5 Data, II

TREC 5: Combining the worst \( i \) systems in order.

input system \( i \) vs. Avg precision

Input retrieval systems sorted worst to best
Experiments

- Randomly choose $n$ input systems.
- For each query:
  - combine, trim, calculate avg precision.
- Calculate mean avg precision.
- Note best input system.
- Repeat (statistical significance).
CombMNZ on TREC3

TREC 3: avg precision over 200 random sets of systems.
CombMNZ on TREC5

TREC 5: avg precision over 200 random sets of systems.
CombMNZ on Vogt

TREC 5 subset: avg precision over between 1 and 200 random sets of systems.
CombMNZ on TREC9

TREC 9: avg precision over 200 random sets of systems.

- CombSUM
- CombMNZ
- The best input system
Metasearch via Voting

[Aslam, Montague]

- Analog to *election strategies.*
  - Requires only rank information.
  - No training required.
Classes of Metasearch Problems

<table>
<thead>
<tr>
<th>ranks only</th>
<th>no training data</th>
<th>training data</th>
</tr>
</thead>
<tbody>
<tr>
<td>relevance scores</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Borda, Condorcet, rCombMNZ</td>
<td>Bayes</td>
<td></td>
</tr>
<tr>
<td>CombMNZ</td>
<td>LC model</td>
<td></td>
</tr>
</tbody>
</table>
Election Strategies

- Plurality vote.
- Approval vote.
- Run-off.
- Preferential rankings:
  - instant run-off,
  - Borda count (positional),
  - Condorcet method (head-to-head).
Metasearch Analogy

- Documents are *candidates*.
- Systems are *voters* expressing preferential rankings among candidates.
Borda Count

- Consider an $n$ candidate election.
- One method for choosing winner is the Borda count. [Borda, Saari]
  - For each voter $i$
    - Assign $n$ points to top candidate.
    - Assign $n-1$ points to next candidate.
    - ...
  - Rank candidates according to point sum.
Election 2000: Florida

### FLORIDA VOTE COUNT

<table>
<thead>
<tr>
<th></th>
<th>Nov. 7</th>
<th>Recount</th>
<th>Certified</th>
<th>12/8 Ruling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bush</td>
<td>1,725</td>
<td>930</td>
<td>537</td>
<td>193</td>
</tr>
</tbody>
</table>

Source: Associated Press

### PRESIDENT DEC. 13

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Votes</th>
<th>Vote %</th>
<th>States Won</th>
<th>EV</th>
</tr>
</thead>
<tbody>
<tr>
<td>R Bush</td>
<td>2,909,176</td>
<td>49 %</td>
<td>29</td>
<td>0</td>
</tr>
<tr>
<td>D Gore</td>
<td>2,907,451</td>
<td>49 %</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>G Nader</td>
<td>96,837</td>
<td>2 %</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I Browne</td>
<td>18,856</td>
<td>0 %</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RF Buchanan</td>
<td>17,356</td>
<td>0 %</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I Phillips</td>
<td>4,280</td>
<td>0 %</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I Hagelin</td>
<td>2,287</td>
<td>0 %</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

25 electoral votes at stake

winner declared

results as of 5:46 p.m. EST
Borda Count: Election 2000

- Ideological order: Nader, Gore, Bush.
- Ideological voting:
  - Nader voter: Nader, Gore, Bush.
  - Gore voter:
    - Gore, Bush, Nader.
    - Gore, Nader, Bush.

\[
\begin{align*}
\text{50/50, 100/0}
\end{align*}
\]
Election 2000: Ideological Florida Voting

<table>
<thead>
<tr>
<th></th>
<th>Gore</th>
<th>Bush</th>
<th>Nader</th>
</tr>
</thead>
<tbody>
<tr>
<td>50/50</td>
<td>14,734,379</td>
<td>13,185,542</td>
<td>7,560,864</td>
</tr>
<tr>
<td>100/0</td>
<td>14,734,379</td>
<td>14,639,267</td>
<td>6,107,138</td>
</tr>
</tbody>
</table>

Gore Wins
Borda Count: Election 2000

- Ideological order: Nader, Gore, Bush.
- Manipulative voting:
  - Gore voter: Gore, Nader, Bush.
  - Nader voter: Nader, Gore, Bush.
# Election 2000: Manipulative Florida Voting

<table>
<thead>
<tr>
<th>Gore</th>
<th>Bush</th>
<th>Nader</th>
</tr>
</thead>
<tbody>
<tr>
<td>11,825,203</td>
<td>11,731,816</td>
<td>11,923,765</td>
</tr>
</tbody>
</table>

Nader Wins
Metasearch via Borda Counts

- Metasearch analogy:
  - Documents are *candidates*.
  - Systems are *voters* providing preferential rankings.

- Issues:
  - Systems may rank different document sets.
  - How to deal with unranked documents?
Borda on TREC5 Data, I

TREC 5: Combining the top i systems in order.

- First i input systems combined by Borda-fuse
- First i input systems combined by CombMNZ
- Input system i

Avg. precision vs. Input retrieval systems sorted best to worst
Borda on TREC5 Data, II

TREC 5: Combining the worst i systems in order.

- first i input systems combined by Borda-fuse
- first i input systems combined by CombMNZ
- input system i
Borda on TREC5 Data, III

TREC 5: Avg precision over random systems.

Number of random input systems

Avg precision

Borda-fuse
CombMNZ
max
avg
min
Condorcet Voting

- Each ballot ranks all candidates.
- Simulate head-to-head run-off between each pair of candidates.
- Condorcet winner: candidate that beats all other candidates, head-to-head.
Election 2000: Florida

<table>
<thead>
<tr>
<th>FLORIDA VOTE COUNT</th>
<th>PRESIDENT DEC. 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov. 7</td>
<td>Recount</td>
</tr>
<tr>
<td>Bush √</td>
<td>1,725</td>
</tr>
<tr>
<td>D Gore</td>
<td>2,907,451</td>
</tr>
<tr>
<td>G Nader</td>
<td>96,837</td>
</tr>
<tr>
<td>I Browne</td>
<td>18,856</td>
</tr>
<tr>
<td>RF Buchanan</td>
<td>17,356</td>
</tr>
<tr>
<td>I Phillips</td>
<td>4,280</td>
</tr>
<tr>
<td>I Hagelin</td>
<td>2,287</td>
</tr>
</tbody>
</table>

25 electoral votes at stake

winner declared by 12/8 Ruling
Condorcet Paradox

- Voter 1: A, B, C
- Voter 2: B, C, A
- Voter 3: C, A, B

Cyclic preferences: cycle in Condorcet graph.

Condorcet consistent path: Hamiltonian.

For metasearch: any CC path will do.
Condorcet Consistent Path
Hamiltonian Path Proof

Base Case:

Inductive Step:
Condorcet-fuse: Sorting

- Insertion-sort suggested by proof.
- Quicksort too; $O(n \log n)$ comparisons.
  - $n$ documents.
- Each comparison: $O(m)$.
  - $m$ input systems.
- Total: $O(m n \log n)$.
- Need not compute entire graph.
Condorcet-fuse on TREC3

TREC 3: avg precision over 200 random sets of systems.
Condorcet-fuse on TREC5

TREC 5: avg precision over 200 random sets of systems.
Condorcet-fuse on Vogt

TREC 5 subset: avg precision over between 1 and 200 random sets of systems.

- CombMNZ (relevance scores simulated with ranks, unrel: 0)
- Quicksort Condorcet
Condorcet-fuse on TREC9

TREC 9: avg precision over 200 random sets of systems.
Outline

- Introduce problem
- Characterize problem
- Survey techniques
  - Upper bounds for metasearch
Upper Bounds on Metasearch

- How good can metasearch be?
- Are there fundamental limits that methods are approaching?
Upper Bounds on Metasearch

- Constrained oracle model:
  - omniscient metasearch oracle,
  - constraints placed on oracle that any reasonable metasearch technique must obey.

- What are “reasonable” constraints?
Naïve Constraint

- *Naïve* constraint:
  - Oracle may only return docs from underlying lists.
  - Oracle may return these docs in any order.
  - Omniscient oracle will return relevant docs above irrelevant docs.
TREC5: Naïve Bound
Pareto Constraint

- *Pareto* constraint:
  - Oracle may only return docs from underlying lists.
  - Oracle must respect *unanimous* will of underlying systems.
  - Omniscient oracle will return relevant docs above irrelevant docs, subject to the above constraint.
TREC5: Pareto Bound

TREC 5: avg precision over 200 random sets of systems.

- Naive Bound
- Pareto Bound
- Condorcet-fuse
- The best input system

Number of randomly chosen input systems

Avg precision
Majoritarian Constraint

- *Majoritarian* constraint:
  - Oracle may only return docs from underlying lists.
  - Oracle must respect *majority* will of underlying systems.
  - Omniscient oracle will return relevant docs above irrelevant docs and break cycles optimally, subject to the above constraint.
TREC5: Majoritarian Bound

TREC 5: avg precision over 200 random sets of systems.
Upper Bounds: TREC3
Upper Bounds: Vogt
Upper Bounds: TREC9

TREC 9: avg precision over 200 random sets of systems.