Authenticating the Query Results of Text Search Engines

H. Pang & K. Mouratidis

VLDB 2008
Motivation

• If a search engine is compromised:
  – Incomplete results: omitting some legitimate documents
  – Altered ranking: deviating from the correct similarity ranking
  – Spurious results: includes fake documents
Contribution:

• A framework for inverted index authentication that can generate an integrity proof for any search result.
Similarity score computation

\[ S(d|Q) = \sum_{t \in Q} w_{Q,t} \times w_{d,t} \]

where

\[ K_d = k_1 \left( (1 - b) + b \frac{W_d}{W_A} \right) \]

\[ w_{d,t} = \frac{(k_1 + 1)f_{d,t}}{K_d + f_{d,t}} \]

\[ w_{Q,t} = \ln \left( \frac{n - f_t + 0.5}{f_t + 0.5} \right) \times f_{Q,t} \]

- Here \( w(d, t) \) and \( w(Q, t) \) are normalized frequency of term \( t \) in document \( d \) and query \( Q \), respectively.
- Execution time proportional to the document size \( n \)
Inverted Index

- Frequency-ordered inverted index.

<table>
<thead>
<tr>
<th>term id</th>
<th>term t</th>
<th>$f_t$</th>
<th>Inverted List for $t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>and</td>
<td>1</td>
<td>(6, 0.159)</td>
</tr>
<tr>
<td>2</td>
<td>big</td>
<td>2</td>
<td>(2, 0.148) (3, 0.088)</td>
</tr>
<tr>
<td>3</td>
<td>dark</td>
<td>1</td>
<td>(6, 0.079)</td>
</tr>
<tr>
<td>4</td>
<td>did</td>
<td>1</td>
<td>(4, 0.125)</td>
</tr>
<tr>
<td>5</td>
<td>gown</td>
<td>1</td>
<td>(2, 0.074)</td>
</tr>
<tr>
<td>6</td>
<td>had</td>
<td>1</td>
<td>(3, 0.088)</td>
</tr>
<tr>
<td>7</td>
<td>house</td>
<td>2</td>
<td>(3, 0.088) (2, 0.074)</td>
</tr>
<tr>
<td>8</td>
<td>in</td>
<td>5</td>
<td>(6, 0.159) (2, 0.148) (5, 0.142) (1, 0.058) (7, 0.058) (8, 0.053) ...</td>
</tr>
<tr>
<td>9</td>
<td>keep</td>
<td>3</td>
<td>(5, 0.088) (1, 0.088) (3, 0.088)</td>
</tr>
<tr>
<td>10</td>
<td>keeper</td>
<td>3</td>
<td>(4, 0.125) (5, 0.088) (1, 0.088)</td>
</tr>
<tr>
<td>11</td>
<td>keeps</td>
<td>3</td>
<td>(5, 0.088) (1, 0.088) (6, 0.079)</td>
</tr>
<tr>
<td>12</td>
<td>light</td>
<td>1</td>
<td>(6, 0.079)</td>
</tr>
<tr>
<td>13</td>
<td>night</td>
<td>3</td>
<td>(5, 0.177) (4, 0.125) (1, 0.088)</td>
</tr>
<tr>
<td>14</td>
<td>old</td>
<td>4</td>
<td>(2, 0.148) (4, 0.125) (1, 0.088) (3, 0.088)</td>
</tr>
<tr>
<td>15</td>
<td>sleeps</td>
<td>1</td>
<td>(6, 0.079)</td>
</tr>
<tr>
<td>16</td>
<td>the</td>
<td>6</td>
<td>(5, 0.265) (3, 0.263) (6, 0.200) (1, 0.159) (2, 0.148) (4, 0.125) ...</td>
</tr>
</tbody>
</table>
PSCAN algorithm

To find the top $r$ matching documents for a query $Q$, using a frequency-ordered inverted index.

(1) Fetch the first $\langle d, w_{d,t} \rangle$ entry in each query term $t$’s inverted list.
(2) While inverted list entries remain,
   (a) Identify the inverted list entry $\langle d, w_{d,t} \rangle$ with the highest term score $c = w_{Q,t} \times w_{d,t}$, breaking ties arbitrarily.
   (b) If $d$ has not been encountered before, create an accumulator $A_d$ and initialize it to zero.
   (c) $A_d \leftarrow A_d + c$.
   (d) Fetch the next entry in term $t$’s inverted list.
(3) Identify the $r$ largest $A_d$ values and return the corresponding documents.
Modified PSCAN algorithm (TRA)

- Fagin’s algorithm (extended)
  - Algorithm repeatedly reads off the next entry \( <d_i, f_i> \) from the inverted list entry \( L_i \), with the largest term score
  - The similarity score of document \( d_i \) is immediately computed (by Random Access) from the document MHT
  - Entry \( <d, s> \) is entered into the result list \( R \) ordered in non-increasing order
  - Threshold \( thres \) is computed which is an upper bound on the similarity score for any non-encountered document down the inverted list
  - So when \( thres \leq R.s_r \) then the top \( r \) matching documents have been found
TRA algorithm

- To prove that R satisfies the correctness criterion
  - For each result document d, the VO includes the query term frequencies in d so that the user can compute the similarity score
  - For each non-result document d’ that occurs up to the cut-off threshold the VO includes the query term frequencies in d’ so that the user can verify that the score is lower than those of the result documents
  - The inverted list entries that correspond to the cut-off threshold
Term MHT

\[ \text{root} = \text{sign}(h(\text{“the”} | f_{t16} | 16 | h_{1-8})) \]
Document MHT

\[
\text{root} = \text{sign}(h(h(\text{doc}_6) \mid 6 \mid h_{1-7}))
\]
Drawbacks

• Search engine has to retrieve entire inverted lists in order to regenerate the complimentary digests of the term-MHTs

• Leaves of the term-MHTs and document-MHTs are smaller than the upper level digests.
Authentication with Chain MHTs

- Digest of each block is included in the digest computation of the block immediately ahead of it.
- Can be used to verify any j leading blocks of the inverted list by supplying the digest of the j+1 block.
Thank you !!!

Questions ?