A Distributed Ranking Algorithm for the iTrust Information Search and Retrieval System

Presented by Boyang Peng

Research conducted in collaboration with
Y. T. Chuang, I. Michel Lombera, L. E. Moser, and P. M. Melliar-Smith
Supported in part by NSF Grant CNS 10-16103
Overview

1) Introduction
2) Overview of iTrust
3) Distributed Ranking System
4) Trustworthiness
5) Evaluation
6) Conclusion and Future Work
Introduction

• What is iTrust?
Introduction

iTrust: Trustworthy Distribution and Retrieval Network

vs

Google, Yahoo, Bing
Purpose

Censorship
1. Distribution of metadata

Source of Information

Source of Information

1. Distribution of metadata
Source of Information

Requester of Information

3. Request encounters metadata

2. Distribution of Requests

3. Request encounters metadata

Source of Information
Source of Information

4. Request matched

Requester of Information

Source of Information

4. Request matched
Distributed Ranking System

• Why ranking is needed in iTrust?
  ▪ Centralized search engines have this functionality
  ▪ Filters out trivial and not-relevant files
  ▪ Increases both the fidelity and the quality of the results

• How is the ranking done?
  ▪ What metrics the ranking algorithm will use and what the ranking formula is.
  ▪ What information the ranking algorithm needs and how to retrieve that information
Distributed Ranking System

• Indexing performed at the source nodes
  ▪ Generate a term-frequency table for an uploaded document

• Ranking performed at the requesting node
  ▪ Ensure fidelity of results
5. Retrieve term-frequency table

Requester of Information

<table>
<thead>
<tr>
<th>Term</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>for</td>
<td>25</td>
</tr>
<tr>
<td>of</td>
<td>18</td>
</tr>
<tr>
<td>said</td>
<td>11</td>
</tr>
<tr>
<td>reach</td>
<td>10</td>
</tr>
<tr>
<td>belly</td>
<td>4</td>
</tr>
</tbody>
</table>

Source of Information

<table>
<thead>
<tr>
<th>Term</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>20</td>
</tr>
<tr>
<td>and</td>
<td>15</td>
</tr>
<tr>
<td>request</td>
<td>10</td>
</tr>
<tr>
<td>retro</td>
<td>9</td>
</tr>
<tr>
<td>believe</td>
<td>8</td>
</tr>
</tbody>
</table>
Ranking Algorithm

\[
Score(d, q) = norm(d) \times \sum_{t \in q} tf(t, d) \times idf(t)
\]

where

- \( norm(d) \) is the normalization factor for document \( d \), computed as:
  \[
  \log \left( 1 + \frac{\text{number\_of\_uncommon\_terms}(d)}{1 + \text{number\_of\_common\_terms}(d)} \right)
  \]

- \( \text{number\_of\_common\_terms} \) for a document \( d \) is \( |s \cap c| \), where \( s \) is the set of all terms in the \( \text{freqTable}(d) \) and \( c \) is the set of common terms

- \( \text{number\_of\_uncommon\_terms} \) for a document \( d \) is \( |\text{freqTable}(d)| - \text{number\_of\_common\_terms} \)
Ranking Algorithm

\[ \text{Score}(d, q) = \text{norm}(d) \times \sum_{t \in q} \text{tf}(t, d) \times \text{idf}(t) \]

where

- \( \text{tf}(t, d) \) is the term-frequency factor for term \( t \) in document \( d \), computed as:
  \[
  \log \left( \frac{1 + \text{freq}(t, d)}{\log(1 + \text{avg}(\text{freq}(d)))} \right)
  \]

- \( \text{freq}(t, d) \) is the frequency of occurrence of term \( t \) in \( \text{freqTable}(d) \)

- \( \text{avg}(\text{freq}(d)) \) is the average frequency of terms contained in the \( \text{freqTable}(d) \)
Ranking Algorithm

\[ \text{Score}(d, q) = \text{norm}(d) \times \sum_{t \in q} \text{tf}(t, d) \times \text{idf}(t) \]

where

- \( \text{idf}(t) \) is the inverse document frequency factor for term \( t \), computed as:

\[ 1 + \log \left( \frac{\text{numDocs}}{\text{docFreq}(t) + 1} \right) \]

- \( \text{numDocs} \) is the total number of documents being ranked
- \( \text{docFreq}(t) \) is the number of documents that contain the term \( t \)
Trustworthiness

• Potential scammers
  ▪ Falsifying Information
    • Distribute a term-frequency table containing every single word in the language
    • Set a limit on the size of term-frequency table of a document
  ▪ Exaggerating Information
    • A malicious node can exaggerate the information about a document to achieve a higher ranking
The percent time that a document is ranked last as a function of the number of keywords in the query. The size of the term-frequency table for all documents is 200.
The mean score of 1000 rankings of a document as a function of the number of keywords in the query. The lines, Document x 5 and Document x 10, correspond to the frequencies in the term-frequency table of a document multiplied by a factor of 5 and 10, respectively.
Evaluation

• Because iTrust is a distributed and probabilistic system, for reproducibility of results, we evaluate the effectiveness of the ranking system by simulation, separate from the iTrust system implementation.

• As the number of keywords in the query increases, the accuracy of the results increases.
The mean percent time of 1000 rankings that a set of documents (Document Set 1 at the left and Document Set 2 at the right) are ranked the top four, as a function of the number of keywords in the query.
Ranking Stability

1 to 5 keywords in query

6 to 10 keywords in query
Ranking Stability

Mean number of position changes vs. Number of keywords in query

WEBIST 2013
iTrust
Boyang Peng
How big should the term-frequency tables be?

![Graph showing the mean number of position changes against the size of term-frequency tables. The graph indicates a U-shaped curve with a minimum at around 200 and a maximum at around 500.](image_url)
Conclusion

- We have presented a Distributed Ranking System for iTrust
  - Effective in ranking documents relative to their relevance to queries that the user has input
  - Exhibits stability in ranking documents
  - Counter scams by malicious nodes
Related Works


Future Work

• Ranking that also takes into account the reputation of the source node or the document or both
• Evaluating the distributed ranking algorithm on a larger and more varied set of documents
• Additional schemes to prevent malicious nodes from gaining an unfair advantage
Our iTrust Website:
- http://itrust.ece.ucsb.edu

Contact information:
- Boyang Peng: bpeng@umail.ucsb.edu
- Personal Website: http://www.cs.ucsb.edu/~jpeng/

Our project is supported by NSF: CNS 10-16193