A Recommendation System for Software Function Discovery

Naoki Ohsugi
Software Engineering Laboratory,
Graduate School of Information Science,
Nara Institute of Science and Technology

Tuesday 16 December, 2003.
International Workshop on Community-Driven Evolution of Knowledge Artifacts
Growth of Software Functions

- Application software is getting more complicated and providing more functions.
  - Total number of menu items (Microsoft Office)
    - Word 2000: 660
    - Word 2002: 772
    - Excel 2000: 705
    - Excel 2002: 792
    - PowerPoint 2000: 565
    - PowerPoint 2002: 646

Users can’t find useful functions from too many functions.
Users Could Not Find Some Useful Functions!

Subjects: 32 users in our lab.
Period: 22 months

- **Excel 2000**
  - Total: 705 functions
  - Maximum: 565 functions
  - Minimum: 189 functions
  - Average: 10.6% of 705

- **Excel 2002**
  - Total: 792 functions
  - Maximum: 83 functions
  - Minimum: 12 functions
  - Average: 10.5% of 792

- **PPT 2000**
  - Total: 646 functions
  - Maximum: 31 functions
  - Minimum: 80 functions
  - Average: 33.5% of 646

- **PPT 2002**
  - Total: 660 functions
  - Maximum: 67 functions
  - Minimum: 22 functions
  - Average: 22.8% of 660

- **Word 2000**
  - Total: 772 functions
  - Maximum: 120 functions
  - Minimum: 11 functions
  - Average: 15.5% of 772

- **Word 2002**
  - Total: 43 functions
  - Maximum: 32 functions
  - Minimum: 11 functions
  - Average: 4.1% of 43
A Recommendation System for Software Function Discovery

- The system recommends individual users a set of candidate functions, which may be useful.
- Our solution is a Collaborative Filtering approach.

Here’s my recommendation:

- **Tools** → **Word Count**… 21 pts
- **Insert** → **Date Time**… 20 pts
- **Tools** → **Thesaurus**… 18 pts
- **Insert** → **Footnote**… 18 pts
- **Tools** → **Spelling**… 17 pts
What is Collaborative Filtering (CF)?

- "Collaborative" means using some users’ knowledge for filtering.
- "Filtering" means selecting useful items from large amount of items.
Voting-based Recommendation Systems with CF

- The systems collect *explicit* votes as users’ knowledge.

Amazon.com
(Book recommendation system)
http://www.amazon.com

MovieLens
(Movie recommendation system)
http://www.movielens.umn.edu
Logging Usage as Users’ Knowledge

- The proposed system automatically collects the records of executed functions (Usage logs) as users' knowledge.

- Usage logs are collected from some users via the Internet.

Usage log as shown below:
2002/02/03 18:50:41 Formatting->Font…
2002/02/03 18:50:45 File->Save As…
Step 1: Computing Similarities

- Computing similarities between the target user and the other users

User 1  Target user  User 2  User 3  User 4

Function A  Function A  Function A  Function E  Function H
Function B  Function A  Function B  Function F  Function I
Function C  Function C  Function C  Function G  Function J
Function D  Function D  Function D

Similar users  Dissimilar users
Step 2: Delivering Knowledge

- Delivering the useful functions candidate, which were frequently used by the similar users'.
Conventional Similarity Calculation

- Calculating Similarities by Correlation Coefficient
  - The dominant frequencies (e.g., “Undo” or “Save”) over-affect similarity computations.

<table>
<thead>
<tr>
<th>Target user</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Undo</td>
<td>60%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Save</td>
<td>20%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Redo</td>
<td>10%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Copy</td>
<td>4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Paste</td>
<td>3%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Cut</td>
<td>2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Clear</td>
<td>1%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User 2</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Save</td>
<td>55%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Undo</td>
<td>25%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Redo</td>
<td>10%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Copy</td>
<td>4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Paste</td>
<td>3%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Cut</td>
<td>2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Clear</td>
<td>1%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User 3</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Undo</td>
<td>60%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Save</td>
<td>20%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Clear</td>
<td>6%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Cut</td>
<td>5%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Copy</td>
<td>4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Paste</td>
<td>3%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Redo</td>
<td>2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Correlation based similarity: +0.41, +0.97

(Range of value [-1.00, +1.00])
Better Similarity Calculation

- Calculating Similarities by Rank Correlation
  - The dominant frequencies ("Undo" & "Save") do not affect similarity computations.

<table>
<thead>
<tr>
<th>Target user</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Undo</td>
<td>1 Save</td>
<td>1 Undo</td>
</tr>
<tr>
<td>2 Save</td>
<td>2 Undo</td>
<td>2 Save</td>
</tr>
<tr>
<td>3 Redo</td>
<td>3 Redo</td>
<td>3 Clear</td>
</tr>
<tr>
<td>4 Copy</td>
<td>4 Copy</td>
<td>4 Cut</td>
</tr>
<tr>
<td>5 Paste</td>
<td>5 Paste</td>
<td>5 Copy</td>
</tr>
<tr>
<td>6 Cut</td>
<td>6 Cut</td>
<td>6 Paste</td>
</tr>
<tr>
<td>7 Clear</td>
<td>7 Clear</td>
<td>7 Redo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undo</td>
<td>60%</td>
<td>55%</td>
<td>60%</td>
</tr>
<tr>
<td>Save</td>
<td>20%</td>
<td>25%</td>
<td>20%</td>
</tr>
<tr>
<td>Redo</td>
<td>10%</td>
<td>10%</td>
<td>6%</td>
</tr>
<tr>
<td>Copy</td>
<td>4%</td>
<td>4%</td>
<td>5%</td>
</tr>
<tr>
<td>Paste</td>
<td>3%</td>
<td>3%</td>
<td>4%</td>
</tr>
<tr>
<td>Cut</td>
<td>2%</td>
<td>2%</td>
<td>3%</td>
</tr>
<tr>
<td>Clear</td>
<td>1%</td>
<td>1%</td>
<td>2%</td>
</tr>
</tbody>
</table>

Correlation based similarity  +0.41  +0.97
Rank correlation based similarity  +0.90  +0.05
Evaluating Accuracy of Recommendation

- Yao’s \textit{ndpm} measure

\begin{itemize}
  \item Interview for user
  \item Usage logs
    \begin{itemize}
      \item 6 users
      \item 22 months
    \end{itemize}
  \item System
  \item \textit{Ndpm} \[0.0, 1.0]\]
    \begin{itemize}
      \item 0.0 is the best
      \item 1.0 is the worst
    \end{itemize}
\end{itemize}
Experimental Result

Collected usage logs of Ms-Word 2000
Subjects: 6 users in our lab.
Period: 22 months

Each user’s ndpm
Average of ndpm
0.5 of ndpm

Algorithms
Random
User Count
Base Case
Correlation based Similarity
Rank Correlation based Similarity

Ndpm

0.6

0.5

0.4

0.3

0.2

0.514
0.404
0.396
0.383
0.355
Conclusion

- I proposed a recommendation system to help users discover useful functions.

- I evaluated the accuracy of recommendation.
  - The result suggested the proposed system has a potential to provide useful recommendation for software function discovery.