DESIGN AND IMPLEMENTATION OF A WEB MINING RESEARCH SUPPORT SYSTEM

Jin Xu

Department of Computer Science & Engineering
Presented at the Ph. D. Candidacy exam

Advisors: Dr. Gregory Madey
Dr. Patrick Flynn

Nov. 21, 2003
INTRODUCTION

- World Wide Web
  - Abundant information
- Web resources
  - On-line databases
  - Web documents
  - Archives: forums, newsgroups
- Important resource for research
OPEN SOURCE SOFTWARE STUDY

- Open Source Software (OSS) study at ND
  - Development of projects
  - Behaviors of developers

- Web resources
  - SourceForge Developer Site
    - Largest OSS development site
      - 70,000 projects, 90,000 developers, 700,000 users
    - Detailed information
      - Project characteristics
      - Developer activities

Jin Xu, CSE Dept.
MOTIVATION

- Traditional Web Data Retrieval
  - Browsing: following links
  - keyword query: irrelevant results
- Web data features
  - Semi-structured
  - Heterogeneous
  - Dynamic
- Web data mining research support systems
  - Web-based IR support system, J.T. Yao, et al, 2003
SYSTEM DESCRIPTION

- **Objective**
  - General solution to use web resources and discover knowledge

- **Functions**
  - Web resource identification – IR
  - Web data extraction – IE
  - Data mining – Generalization
  - Analysis & Validation

- Will be tested on the OSS study at ND
FRAMEWORK

Web Mining Research Support System

- Information Retrieval
- Information Extraction
- Generalization
- Simulation & Validation

Open Source Software
WEB INFORMATION RETRIEVAL

- Current IR tools
- Web page classification
- Proposed classification search tool
- Example
CURRENT IR TOOLS

- Current IR Tools
  - Directory - subject lists
  - Search engine - keywords query
- Limitations
  - Low precision - too many irrelevant documents
  - Low recall - too little of the web is covered by well-categorized directories
  - Not well-categorized for research needs
- Classification search tool
WEB PAGE CLASSIFICATION

- Manual classification
  - Accurate
  - Yahoo, Informine
  - Impossible for large number of web pages

- Automatic classification
  - Nearest neighbor (NN) classifier (C. Chekuri, 1996)
    - Similar documents are assigned the same class label
  - Feature selection (S. Chakrabarti, 1998)
    - Eliminate features with low correlations
  - Naïve Bayes classifier (A.K. McCallum, 2000)
CLASSIFICATION SEARCH TOOL

User Interface → Crawlers → Existing search tools

Classier

XML directories → Crawlers

Web pages
EXAMPLE

Hosted sites
www.sourceforge.com
savannah.gnu.org
...

Projects
www.linux.org
www.apache.org
...

Research
Organizations
www.nd.edu/~oss/
...

Papers
...

Conferences
...
WEB INFORMATION EXTRACTION

- Wrapper
- Wrapper generation
- OSS wrapper & data collection
- Wrapper generation tools
- Hybrid IE
WEB INFORMATION EXTRACTION

- **Web IE**
  - Extract a specific portion of web documents
  - Port into databases

- **Wrapper**
  - Extract information from a particular web site
  - Extraction rules and code to apply rules
WRAPPER GENERATION (WG)

- Manual generation
  - Understanding of web documents
  - Writing codes
  - Dynamic changes of web sites
- Semi-automatic generation
  - Sample pages
  - Demonstration by users
- Automatic generation
  - Learn extraction rules
OSS WEB WRAPPER

- OSS web wrapper
  - Perl and CPAN modules
  - URL accessing – fetch pages (LWP)
  - HTML parser – parse pages
    - HTML::TableExtract – extract information
    - Link extractor – extract links
    - Word extractor

- Features
  - Manual
  - High maintenance cost
  - Not suitable to handle free text
WEB DOCUMENT TYPES

- Structured text
  - Fixed order
  - Labels/tags
  - On-line data generated by database
- Free text
  - Natural language text
- Semistructured text

Bill Smith – admin, Perl, Graduate student, University of Notre Dame, IN 46556, (574)333-3333
WG TECHNIQUES

- **Wrapper induction** (SoftMealy, 1998; WIEN, 2000; STALKER 2001; LIXTO 2003)
  - Training examples
  - Extraction rules
  - Format features to delineate the structure of data
  - Not suitable for free text

- **Natural language processing (NLP)** (WHISK, 1999)
  - Training examples
  - Extraction rules
  - Linguistic constraints
  - Work for all
  - Complex, time cost
HYBRID IE ARCHITECTURE

User -> Structured/Semi-Structured Text Parser

Structured/Semi-Structured Text Parsing

Web Document Selector

Selector to Structured/Semi-Structured Text Parser

Structured/Semi-Structured Text to Parse Tree

Parse Tree to Wrapper Induction

Wrapper Induction to Heuristics

Heuristics to Structured Data Extractor

Structured Data to Database

Database to NLP Learning

NLP Learning to Sample Pattern

Sample Pattern to Extraction Rules

Extraction Rules to Sample Pattern
OSS APPLICATION

- Web sites
  - SourceForge, Savannah, Linux, Apache, Mozilla
- Structured/semi-structured
  - Membership tables, statistics tables, etc.
- Free text
  - Message board, emails, etc.
GENERALIZATION

- Overview
- Preprocessing
- Data mining functions
- Previous OSS generalization study
- Infrastructure
GENERALIZATION OVERVIEW

- Discover information patterns
- Two steps
  - Preprocessing
    - Missing, erroneous data
    - Wrong formats
    - Unnecessary characters
  - Pattern recognition
    - Advanced techniques
PREPROCESSING

- **Data cleansing**
  - Eliminate irrelevant/unnecessary items
  - Detect errors/inconsistencies/duplications

- **User identification**
  - A single user uses multiple IP addresses
  - An IP address is used by multiple users

- **Session identification**
  - Divides page accesses of a single user into sessions
DATA MINING FUNCTIONS

- Association Rules (C. Lin, 2000; B. Mobasher 2000)
  - Find interesting association or correlation relationship among data items

- Clustering (Y. Fu, 1999; B. Mobasher 2000)
  - Find natural groups of data
  - Usage clusters – users with similar browsing patterns
  - Page clusters – pages having related content
DATA MINING FUNCTIONS (Cont.)

● Classification (B. Mobasher 2000)
  - Map a data item into predefined classes
  - Develop a profile of items

● Sequential patterns (H. Pinto, 2001)
  - Find patterns related to time or other sequence
  - Prediction
PREVIOUS OSS GENERALIZATION

- Association Rules
  - “all tracks”, “downloads” and “CVS” are associated

- Classification
  - Predict “downloads”
  - Naïve Bayes – Build Time 30 sec, accuracy 9%
  - Adaptive Bayes Network - Build Time 20 min, accuracy 63%

- Clustering
  - K-means
  - O-cluster

- In collaboration with Y. Huang

Jin Xu, CSE Dept.
GENERALIZATION INFRASTRUCTURE

Cleansing
Statistics
Forums
Server logs

Integration
Statistics
Forums
Server logs

Transformation
Integrated data

Reduction
Transformed data

Recognition
Reduced data

Association Rules
Clustering
Classification
Sequential patterns

Jin Xu, CSE Dept.
GENERALIZATION TECHNIQUES

- **Data cleansing**
  - Missing data replaced by mean or similar input
  - Outliers can be detected by distribution, regression

- **Data integration**
  - Metadata or correlation analysis

- **Data transformation**
  - Divide by $10^n$
  - Min-Max
  - Z-score transformation
GENERALIZATION TECHNIQUES (Cont.)

- Data reduction
  - Data aggregation
  - Data compression
  - Discretization

- Pattern discovery
  - Apply data mining functions
OSS APPLICATION

- Patterns to characterize the activeness of a project
  - Downloads, page view, bug reports
- Clustering and dependencies of OSS projects
- Groups of developers and their relationships
SIMULATION & VALIDATION

- Introduction
- Validation approaches
- Previous validation – OSS docking
- Future work
SIMULATION & VALIDATION

- Simulation (In collaboration with Y. Gao)
  - Interpret the mined patterns
  - Build models and simulations
  - Use simulations to test and evaluate hypothesis

- Validation

- OSS validation – docking
VALIDATION

- Three methods of Validation
  - Comparison with real phenomenon
  - Comparison with mathematical models
  - Docking with other simulations

- Docking
  - Verify simulation correctness
  - Discover pros & cons of toolkits
  - R. Axtell, 1996; M. North 2001; M. Ashworth, 2002
OSS DOCKING EXPERIMENT

- **Four Models of OSS**
  - random graphs
  - preferential attachment
  - preferential attachment with constant fitness
  - preferential attachment with dynamic fitness

- **Agent-based Simulation**
  - Swarm
  - Repast
OSS NETWORK

- A classic example of a dynamic social network
- Two Entities: developer, project
- Graph Representation
  - Node – developers
  - Edge – two developers are participating in the same project
- Activities
  - Create projects
  - Join projects
  - Abandon projects
  - Continue with current projects
OSS MODEL

- **Agent:** developer
- **Each time interval:**
  - Certain number developers generated
  - New developers: create or join
  - Old developers: create, join, abandon, idle
  - Update preference for preferential models
DOCKING PROCESS

Docking

Swarm
Repast

E:R
B: A
B: A: C
B: A: D

Diameter
Degree distribution
Clustering coefficient
Community size

Jin Xu, CSE Dept.
DOCKING PROCEDURE

- Process: comparisons of parameters corresponding models.
- Findings:
  - Different Random Generators
  - Databases creation errors in the original version
  - Different starting time of schedulers
DOCKING PARAMETERS

- **Diameter**
  - Average length of shortest paths between all pairs of vertices

- **Degree distribution**
  - The distribution of degrees throughout a network

- **Clustering coefficient (CC)**
  - $CC_i$: Fraction representing the number of links actually present relative to the total possible number of links among the vertices in its neighborhood.
  - $CC$: average of all $CC_i$ in a network

- **Community size**
DEGREE DISTRIBUTION

- Participated projects vs. Developers for Repast and Swarm.

- Line graphs showing the distribution of developers across participated projects for Repast and Swarm.
CLUSTERING COEFFICIENT

![Graphs showing clustering coefficient over time for Repast and Swarm.](image)
COMMUNITY SIZE DEVELOPMENT

- Swarm developers
- Repast developers
- Swarm projects
- Repast projects

Time period (month) vs. Community size
PROPOSED WORK OF VALIDATION

• Docking
  - Many runs instead of one
  - Statistical analysis
  - More network parameters
    • Average degree
    • Cluster size distribution
    • Fitness and life cycle

• Statistic comparison
CONTRIBUTIONS

- Provide an integrated web mining system to support research – a new tool
- Build a classification retrieval tool to improve precision and recall, as well as meet users’ search requirements
- Implement a hybrid IE tool to extract web data effectively and efficiently
- Create a generalization infrastructure which is suitable for web data mining
- Provide methods to validate OSS simulations
TIME PLAN

- IR system – March 2004
- IE system – July 2004
- Generalization – September 2004
- Validation – December 2004
- Dissertation – During the research
- Complete – May 2005
PAPERS

● Published papers
  - “Multi-Model Docking Experiment of Dynamic Social Network Simulations”, *Agents2003*.
  - “Docking Experiment: Swarm and Repast for Social Network Modeling”, *Swarm2003*.

● Future papers
  - A classification web information retrieval tool
  - A hybrid web information extraction tool
  - Data mining results from OSS study
  - Validation results of OSS simulations
THANK YOU
OSS DATA COLLECTION

- Data sources
  - Statistics, forums

- Project statistics
  - 9 fields – project ID, lifespan, rank, page views, downloads, bugs, support, patches and CVS

- Developer statistics
  - Project ID and developer ID
EXAMPLES OF DATA MINING FUNC.

- **Association rules**
  - 40% of users who accessed the web page with URL/project1, also accessed /project2;

- **Clustering**
  - users of project1 can be grouped as developers and common users.

- **Classification**
  - 50% of users who downloaded software in /product2, were developers of Open Source Software and worked in IT companies.

- **Sequential patterns**
  - if clients who downloaded software in /project1, they also downloaded software in /project2 within 15 days

Jin Xu, CSE Dept.
SOCIAL NETWORK MODEL

- **Graph Representation**
  - **Node/vertex** – Social Agent
  - **Edge/link** – Relationship
  - **Index/degree** - The number of edges connected to a node

- **ER (random) Graph**
  - edges attached in a random process
  - No power law distribution

Jin Xu, CSE Dept.
SOCIAL NETWORK MODEL (Cont.)

- Watts-Strogatz (WS) model
  - include some random reattachment
  - No power law distribution
- Barabasi-Albert (BA) model with preferential attachment
  - Addition of preferential attachment
  - Power law distribution
- BA model with constant fitness
  - Addition of random fitness
- BA model with dynamic fitness
SWARM SIMULATION

- **ModelSwarm**
  - Creates developers
  - Controls the activities of developers in the model
  - Generate a schedule

- **ObserverSwarm**
  - Collects information and draws graphs

- **main**

- **Developer (agent)**
  - Properties: ID, degree, participated projects
  - Methods: daily actions
REPAST SIMULATION

- Model
  - creates and controls the activities of developers
  - collects information and draws graphs
    - Network display
    - Movie
    - snapshot
- Developer (agent)
- Project
- Edge
VALIDATION CONCLUSION

- Same results for both simulations
- Better performance of Repast
- Better display provided by Repast
  - Network display
PROJECT PAGE

Statistics for the past 7 days:

<table>
<thead>
<tr>
<th>Date</th>
<th>Rank</th>
<th>Page Views</th>
<th>DI</th>
<th>Bugs</th>
<th>Support</th>
<th>Patches</th>
<th>All Trkr</th>
<th>Tasks</th>
<th>CVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 Nov 2003</td>
<td>98</td>
<td>217,366</td>
<td>81,382</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17 Nov 2003</td>
<td>102</td>
<td>282,258</td>
<td>90,525</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18 Nov 2003</td>
<td>101</td>
<td>283,268</td>
<td>95,168</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19 Nov 2003</td>
<td>102</td>
<td>283,247</td>
<td>95,168</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20 Nov 2003</td>
<td>102</td>
<td>283,268</td>
<td>95,168</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21 Nov 2003</td>
<td>101</td>
<td>283,247</td>
<td>95,168</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22 Nov 2003</td>
<td>102</td>
<td>283,268</td>
<td>95,168</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Statistics for All Time:

<table>
<thead>
<tr>
<th>Lifespan</th>
<th>Rank</th>
<th>Page Views</th>
<th>DI</th>
<th>Bugs</th>
<th>Support</th>
<th>Patches</th>
<th>All Trkr</th>
<th>Tasks</th>
<th>CVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>356 days</td>
<td>82.3</td>
<td>22,815,627</td>
<td>352</td>
<td>253</td>
<td>115</td>
<td>704</td>
<td>259</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

View Reports: Last 7 Days

- Statistics data is currently updated every 24 hours, on a 24-hour delay (i.e., the most recent stats shown will be from the day (Pacific time zone) which took place 48 hours ago.

- NOTE: The SourceForge.net Site Status page reflects the presence of known issues with the accuracy of recent download statistics. Download statistics shown on this page for recent dates may be inaccurate (they will be corrected in the future), rankings based on these statistics may be skewed.

Jin Xu, CSE Dept.