


Quixote

**An HTML to XML
Mining & Integration Tool**

Christina Yip Chung
Database Lab, Security Lab
University of California at Davis
November, 2000



Agenda

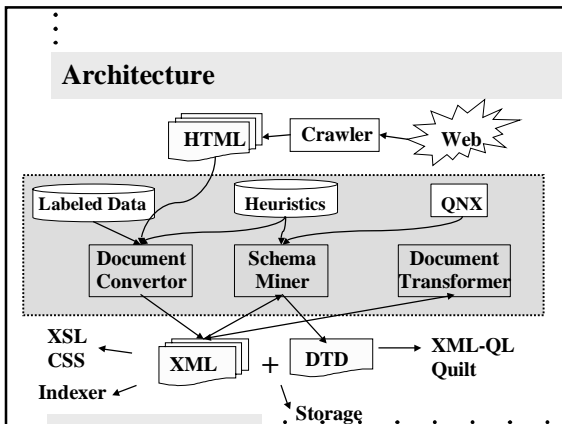
- Introduction
- Step 1: Data Extraction
- Step 2: Schema Discovery
- Step 3: Data Integration
- Other Projects

Motivation

- **Focused crawlers give us a whole bunch of topic-related HTML documents**
 - E.g. IBM's Resume Central Station
 - Resume1 (HTML) Resume2 (HTML)
 - What we want
 - Resume1 (Integrated) Resume2 (Integrated)
- **What are the cool things we can do with them?**
 - Unified, roadmap view on the data
 - Structured queries
 - Storage optimization
 - Indexing

What is Quixote

- **If only these HTML documents were in XML and their schema known** ⇒
- **Quixote, an automated mining and integration tool:**
 - Data Extraction
 - Extract XML tags from text in HTML documents
 - Resume1 (Extracted)
 - Schema Discovery
 - Discover a global approximate schema for the documents
 - Majority Schema
 - Data Transformation
 - Translate the documents to conform to the global schema
 - Resume1 (Transformed)



Goal

- **Input: a collection of topic-specific HTML docs + topic-specific XML tags**
- **Output: a collection of XML docs valid to a DTD**
- **Assumptions on topic-specific documents**
 - <concept, content> : <label, text>
 - No synonyms, Homonyms present
 - Tree hierarchy of concepts
 - Data rich
 - Highly regular
 - Heterogeneous inter-document
 - Homogeneous intra-document

Data Extraction

- **Goal: Extract data records from unstructured / semistructured sources**
- **Related work on mediators**
 - Information Manifold [AT&T] TSIMMIS [Stanford]
 - Garlic [IBM Almaden]
- **Related work on wrappers**
 - TSIMMIS wrapper [Stanford]: manual, regular expression
 - W4F [UPenn]: manual, path expressions
 - YAT [INRIA]: manual, CFG
 - NoDoSE [Adelberg, NW]: learning, text, user-specified schema
 - STALKER [Knoblock, USC]: learning, HTML font + indentation, reg. exp. pattern matching, YACC page specification
 - [Kushmerick, Dublin]: learning, 6 templates, no inferred structure
 - [Embley, Brigham Young]: learning, Ontology, HTML format clues
 - [SUNNY Stony Brook]: learn maximal unambiguous reg. exp.

Data Extraction (2)

- **Classification**
 - Degree of automatic wrapper generation
 - Wrapper language: reg. exp., path exp., CFG, specialized
 - Input data: text/HTML
 - Output data: structured / unstructured
 - Domain knowledge: HTML format clues, HTML tags, HTML tree structure & heuristics, Templates, Ontology
- **Limitations**
 - Manual wrappers: data heterogeneity & dynamics
 - All (except [STALKER, SUNNY]) assume some sort of input 'schema' on data structures
 - Input data: highly homogeneous

Data Extraction (3)

- **Features of our wrapper**
 - Fully automatic
 - Assume no input schema
 - Bayes classifier learns labels
 - Infer structures of labels by
 - HTML format
 - HTML tags
 - HTML tree
 - Constraints on labels: depth, ancestor-descendant, siblings
 - XML documents properties

Data Extraction (3)

Approach

- **User specifies**
 - Topic-specific XML tags (labels)
 - Training data classifying texts to labelsSample training data
- **Bayes classifier associates HTML text to XML tags**
- **Infer structures of labels by reordering heuristics**

Data Extraction (4)

Restructuring rules

- **Heading Rule**
 - Header tags: h1, ..., h6, hr, div
 - Header tags of the same name cannot be nested
- **Phrasing Rule**
 - Phrase tags: header tags, p, tr, ul, ol, dir
 - Everything between phrase tags of the same name are pushed down one level as child nodes
- **Consolidation Rule**
 - HTML tags are replaced by their descendant XML tags
 - List structure maintained for list tags (ul, ol, dir...)
 - Siblings with the same XML tag name are kept as siblings
- **Constraints cannot be violated**

Data Extraction (5)

Evaluation

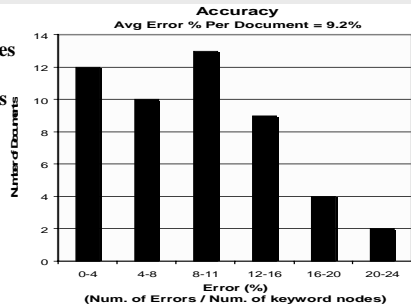
- **Tree traversal: Two traversals for all reordering rules**
- **One Pass Property: Each reordering is applied to a node at most once.**
- **Empirical study**
 - Resumes from IBM's Resume Central Station
 - 25 labels

Data Extraction (6)

Accuracy

- 50 resumes
- Error %

v.s. # of docs

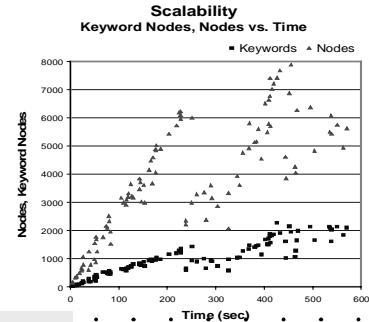


Data Extraction (7)

Scalability

- 100 resumes
- # of nodes

v.s. Time



Data Extraction (8)

Extensions

- **Bayes classifier to identify XML tags**
 - Classification accuracy
 - requires lots of training data
 - restructuring heuristics is sensitive to the accuracy
 - ⇒ Interactive tool for user to give feedback to system
 - Fine tune Bayes classifier: top k-words, words correlation, database + dictionary
- **Reordering heuristics**
 - HTML format: font size, indentation, table
 - repeating HTML tags
- **Rule-based language for reordering heuristics**

Schema Discovery

- **Goal: Infer structures from data**
- **Related work**
 - Dataguide [Stanford]
 - Typing [Stanford]
 - XTRACT [Bell]
 - Graph schema [AT&T]
 - Tree expressions [Singapore]
 - XML view inference [UCSD]
- **Classification**
 - Data models: OEM, DOM
 - Schema models: Graph schema, Dataguide, DTD
 - Precision of schema: Precise, Approximate (less tight)
- **Limitations of related work**
 - OEM: for path queries, not for document structures
 - Precise schema: large schema size
 - Approximate schema: full coverage

Schema Discovery (2)

- **What we want**
 - DOM data model for documents
 - OEM-like model for schemas
 - Approximate schema with appropriate coverage
- **Observation 1**

"Lower coverage in the schema is desirable."

⇒ Propose: Majority schema
- **Observation 2**

"Imprecise modeling reveals regular patterns in data."

⇒ Propose: A data mining guiding principle
- **Observation 3**

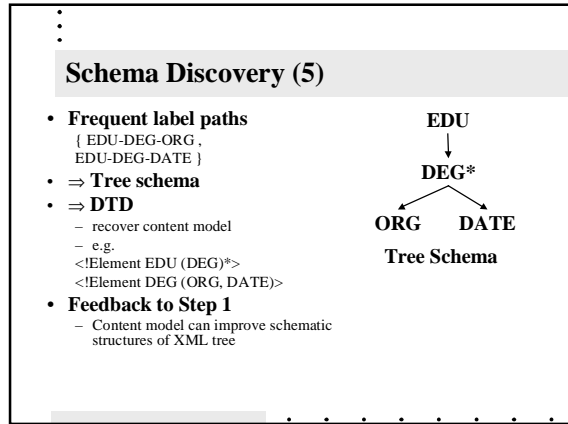
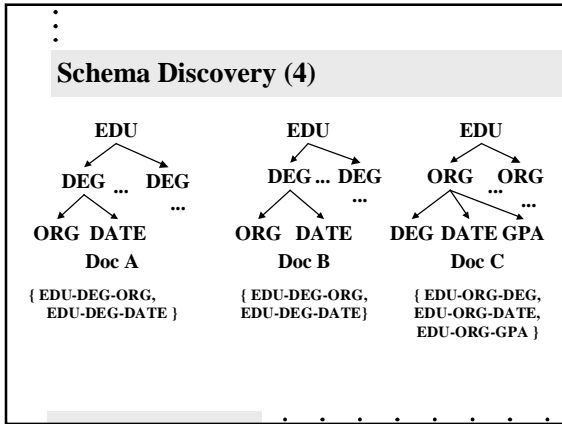
"Domain knowledge available for topic-specific documents"

⇒ Propose: Explicit mechanism to specify a constraints

Schema Discovery (3)

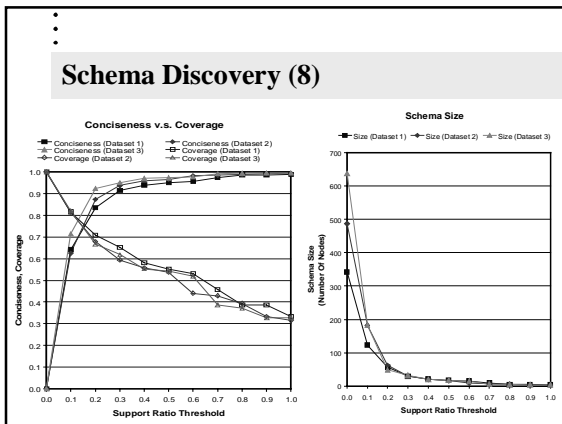
Approach

- **Ignore details in a tree schema**
 - order, multiplicity, grouping, alternative subsequences
- **Tree schema ⇒ a set of label paths.**
 - Aprior algorithm: Mine maximal frequent label paths
- **Maximal frequent label paths ⇒ majority schema**
- **[Optional] Unify similar subtrees in the majority schema**
- **Majority schema ⇒ DTD**
- **Fill in missing details in DTD**
 - order, multiplicity, grouping



- ### Schema Discovery (6)
- **Constraints based Domain Knowledge**
 - Uses
 - Reduce search space
 - Improve accuracy
 - Types
 - Minimum, maximum depth constraints
 - Ancestor-Descendant constraints
 - Sibling constraints
 - Examples
 - Title labels: tags with maximum depth = 2
 - Content labels: tags with minimum depth = 1
 - No label can be the descendant of itself

- ### Schema Discovery (7)
- #### Evaluation
- **Computational complexity**
 - Compute schema (documents X): $O(|E|Deg(X)|X|)$ time
 - Unification (schema S): $O(|S|^2 + ted(S,S)|S|)$ time
 - Feasibility of domain knowledge
 - Exhaustive search space: 8,000,000 nodes
 - Constraints considered: 1900 nodes
 - Pruning added: 70 nodes
 - **Goodness of majority schema**
 - Conciseness: small schema size
 - Coverage: structures in the dataset described by the schema
 - Evaluate majority schemas at different degrees of precision



Schema Discovery (9)

DTDs for 1000 and 50 resumes similar
 \Rightarrow robust to heterogeneity in the documents
 \Rightarrow robust to errors in converting HTML to XML

- **Sample DTD**

```

<!element resume (CONTACT, OBJECTIVE, EDUCATION, EXPERIENCE*, SKILLS, FIELDS?, AWARDS?, ACTIVITIES?, ACHIEVEMENTS?, REFERENCE)*>

<!element CONTACT #PCDATA>
<!element OBJECTIVE #PCDATA>
<!element EDUCATION (#PCDATA, ORGANIZATION*)>
<!element ORGANIZATION (#PCDATA, DATE)>
<!element DATE (#PCDATA, DEGREE)>
<!element EXPERIENCE (#PCDATA, DATE2*)>
<!element DATE2 (#PCDATA, TITLE)>
<!element TITLE (#PCDATA, ORGANIZATION)*>
  
```

Schema Discovery (10)

- **Extensions**
 - Alternative content model
 - Cyclic definitions of elements
 - IDREFs
 - Other schema formalism (XML Schema)

Data Integration

- **Goal: Transform a document to conform to a schema**
- **Related work**
 - [Larson] [Spaccapietra] [Navathe] [Sheth]: schema level
 - [Pu] [Chatterjee] [Li] [Prabhakar]: data level
 - [Gotthard] [Sull] [Bertino]: object-oriented
 - [Milo]: mapping between data models
 - [Murata] [Milo]: type checking
- **Classification**
 - Input data: heterogeneous, same schema in diff. data models
 - Data model: relational, E-R, object-oriented, DTD
 - Manual v.s. automatic
 - Identify related entities
 - data level, schema level (name conflicts, structural conflicts)
 - class hierarchies, methods

Data Integration (2)

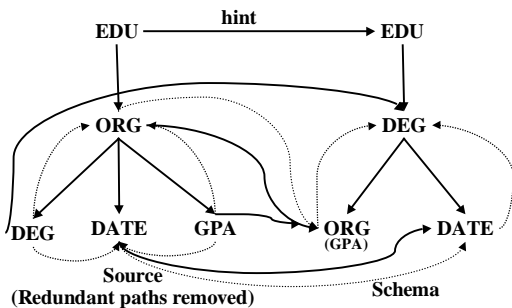
- **Our challenges**
 - Interschema relationship identification
 - not many semantics
 - label names
 - label statistics
 - tree structure
 - Schema integration: fixed global schema
- **Our contribution**
 - Fully automatic
 - XML data
 - Heterogeneous data

Data Integration (3)

Approach

- **Use tree edit distance to map nodes in the data tree to tree schema \Rightarrow 'hint'**
- **Use 'boundary box' to enrich hint to full mapping**
 - unmapped nodes as attributes to its mapped least reference ancestor
 - heuristics to pick among multiple matches that avoid cloning in the transformed tree
- **Construct transformed tree based on the hint**
 - consider content model (multiplicity, grouping)

Data Integration (4)



Data Integration (5)

Extensions

- Transformation rules sensitive to the 'hint'
- Customize tree edit distance costs: support, depth, probability
- Compute 'hint' based on ancestors, siblings, descendants (c.f. approximate Dataguide)
- Noise insensitive
- Evaluation

•
•

Others

- **Other projects**
 - DEMIDS: anomaly detection system
 - MoBy 1-2-3: user profiling system
- **Other research interests**
 - Data mining
 - Web technology
 - Intrusion & anomaly detection systems

• • • • • • • •