UCAI: Challenges and Choices

Digital Library Federation, Fall Forum 2003

Presented by Esme Cowles

Goals

- Ingest: convert three datasets to VRA Core
- Cluster: identify duplicates and create work units
- Prototype: Build a prototype database and interface

Ingest: Formats

- Three formats: MARC, SGML, FoxPro
- XML the logical choice
  - MARC: MARC4J (aka JAMES)
  - SGML: only needed XML declaration
  - FoxPro: exported tables as XML files
- Character encoding
  - MARC: ANSEL
  - SGML/FoxPro: Latin-1

Ingest: Schemas

- Flat v. hierarchical
- Variation within datasets
  - SGML is from a union catalog
  - MARC contains artifacts of previous migration
- Semantic fuzziness
  - MARC less granular than art-specific VRA
  - Noun v. Adjective forms, capitalization
- FoxPro tables not designed for direct access

Ingest: XML Processing

- SAX to break up large XML files into record-level fragments
- XSLT for schema transformation

- Embedded Java function calls for low-level text processing

Ingest: Future Work

- Non-programmer mapping
- Need to handle hierarchy references, precedence rules, formatting, etc.

Clustering: Bottom-Up

- Partitional clustering
- Too many records to compare round-robin
    - Use search engine to get a list of candidates
- False negatives very problematic
    - Start with inclusive measures
    - Use other measures to get better granularity

Clustering: Similarity Measures

- Title
    - Fuzzy matching v. aggressive normalization
    - Thesaurus helpful, but not sustainable
- Creator
    - Fuzzy matching v. last name comparisons
    - Controlled vocabulary
- Date
    - Not present in many records (10-15%)
    - Missing values difficult to resolve in general

Clustering: Top-Down

- Use SGML work units as bins
- Not a total solution because not all records will match
    - These records will need to be clustered independently anyway

Prototype: First Round

- Xindice
    - Open source, great API (xml:db)
    - Slow, buggy, immature
- Tamino
    - Good performance overall, established

    - Problems with large result sets
- TeraText
    - Best performance
    - No Linux version, limited APIs
- Lucene
    - Search engine, not a database
    - Fast, full-featured

Prototype: Lucene + Xindice

- All open source, Java
- Solid performance
- Fatal Flaw: character-encoding bugs

Prototype: Oracle XMLDB

- 9i XML features
- Fulltext searching
- Good performance
- Awkward API (JDBC, SQL, CLOBs)
- High-availability and hybrid possibilities

Last Modified: 2003.11.18