Introduction to Structured Document Clustering & ExtMiner

13.12.2006
Laboratory session
Introduction to Data mining

Miika Nurminen
University of Jyväskylä

Text Mining

- Dörre *et al:* "Text mining applies data mining techniques, such as clustering, classification or association rule search to textual information."
- Typically high-dimensional (or irreducible to a simple vector representation) data
  - E.g. Document term as a dimension
- Text mining is highly dependent on document representation, or index
  - Background knowledge on information retrieval is required
- Data preprocessing (=feature selection) is essential (stopword filtering, term stemming etc)

The baseline: Classical Vector model [Salton]

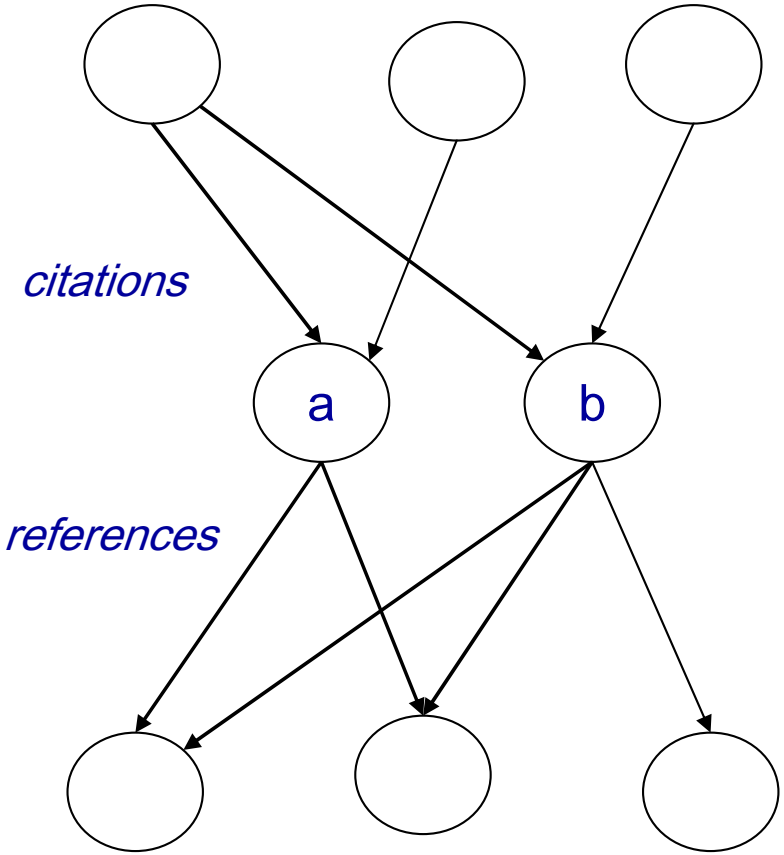
- THE Classical way to represent text documents.
- Documents and queries are represented as vectors. Each index term represents a dimension in document (or query) vector.
- Weighting can be used to emphasize relative importance of a term.
- tf*idf is a popular weighting scheme
  - tf (term frequency) is the relative count of term occurrences in a given document.
  - idf (inverted document frequency) is inverse proportional to number of documents where given term occurs
- Weighting assumption: documents are best described by terms that occur in a minority of documents in a collection (but multiple times documents where they occur in the first place)
- Documents and queries (i.e. ranking) or 2 documents (clustering) can be compared by calculating the dot product between vectors. This is also called cosine similarity.

Structured documents

- Unfortunately, classical vector model is not sufficient for structured documents (HTML, XML, LaTeX etc)
- In addition to text content, documents can have structure, hyperlinks and metadata. Furthermore, text content is dispersed across document structures.
- All these aspects together should be taken to account for effective retrieval – ideally both in indexing and ranking.
- Another problem is document granularity: some documents are logically composed of multiple documents (e.g. chapters or sections in separate file). But a single document can have multiple themes as well (e.g. a single web page containing multiple small news articles).

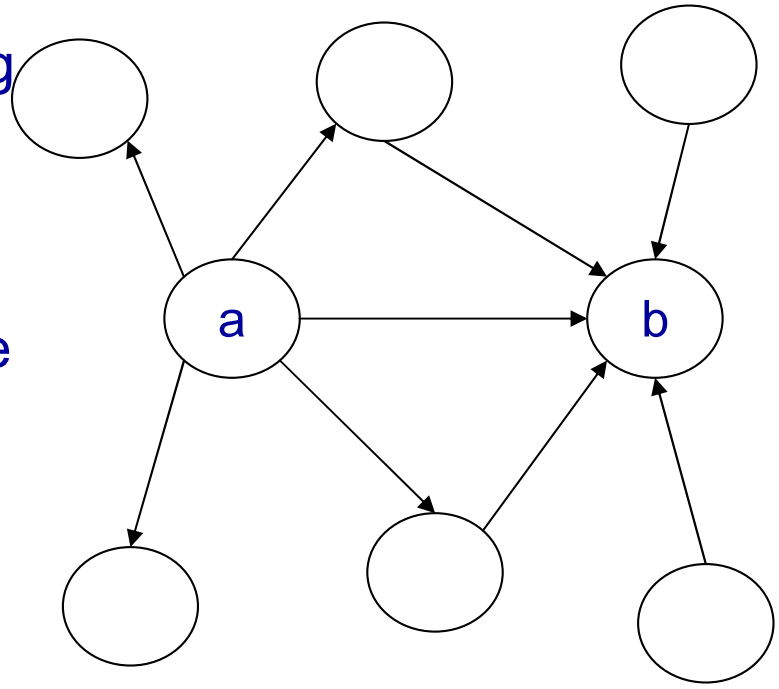
Link-based similarity

- Co-citation and bibiliographic coupling have been used widely for analyzing scientific articles in bibliometrics for decades.
- In the picture, co-citation between a,b is 1/3 (one document that cites both a and b), coupling 2/3 (a and b cite together 2 documents).
- For web documents, co-citation is often preferred over coupling because it evolves as new documents are updated.



Link-based ranking

- Essential algorithms: Pagerank [Brin & Page] & HITS [Kleinberg]
- PageRank resembles the ranking of scientific articles: the more high-ranking pages link to a page, the higher it scores.
- HITS extracts Hub and authority pages from the link graph. Authority pages are approximately same as high-ranking PageRang pages.
- In the picture, a is a hub, b is an authority.
- In addition to link structure, anchor text can be indexed to describe the linked page content (anchor window)



Structured schemes

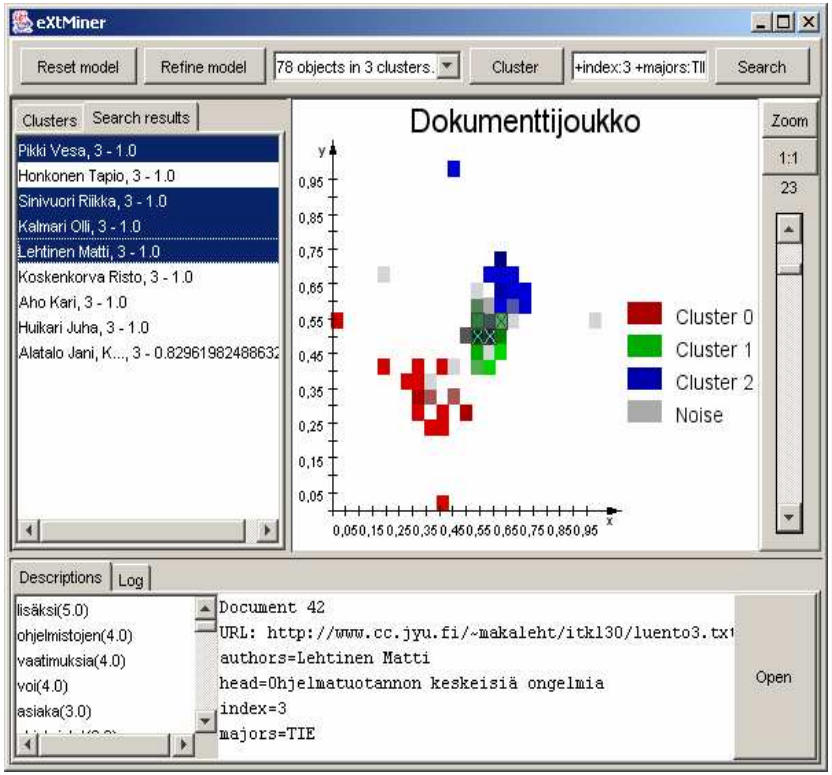
- Interpreting structured document can be interpreted as hypertext document collection: recursive indexing for content (for this special case, traditional vector model can be used) [Frisse]
- Structured searches require complex tree- or graph-based index structures [Weigel]
- Similarity comparison requires graph- or tree matching => slow [Luk]
- In general, vector-based index structures cannot be used effectively with structured documents, however…

Putting aspects together: Extended Vector Model

- …for many practical search schemes it suffices that *selected structures* are extracted from document and indexed as vectors
- or even better: for *metric* clustering a similarity measure will suffice, so document representation is not an issue
- Extended Vector Model [Fox] provides simple and general-purpose mechanism for calculating similarity from multiple components as weighted linear sum. Each component represents a collection of document features that can be calculated separately using classical vector model or other similarity measure.
- For example, there could be a component for document content in selected elements, another for link structures and third for various metadata fields.

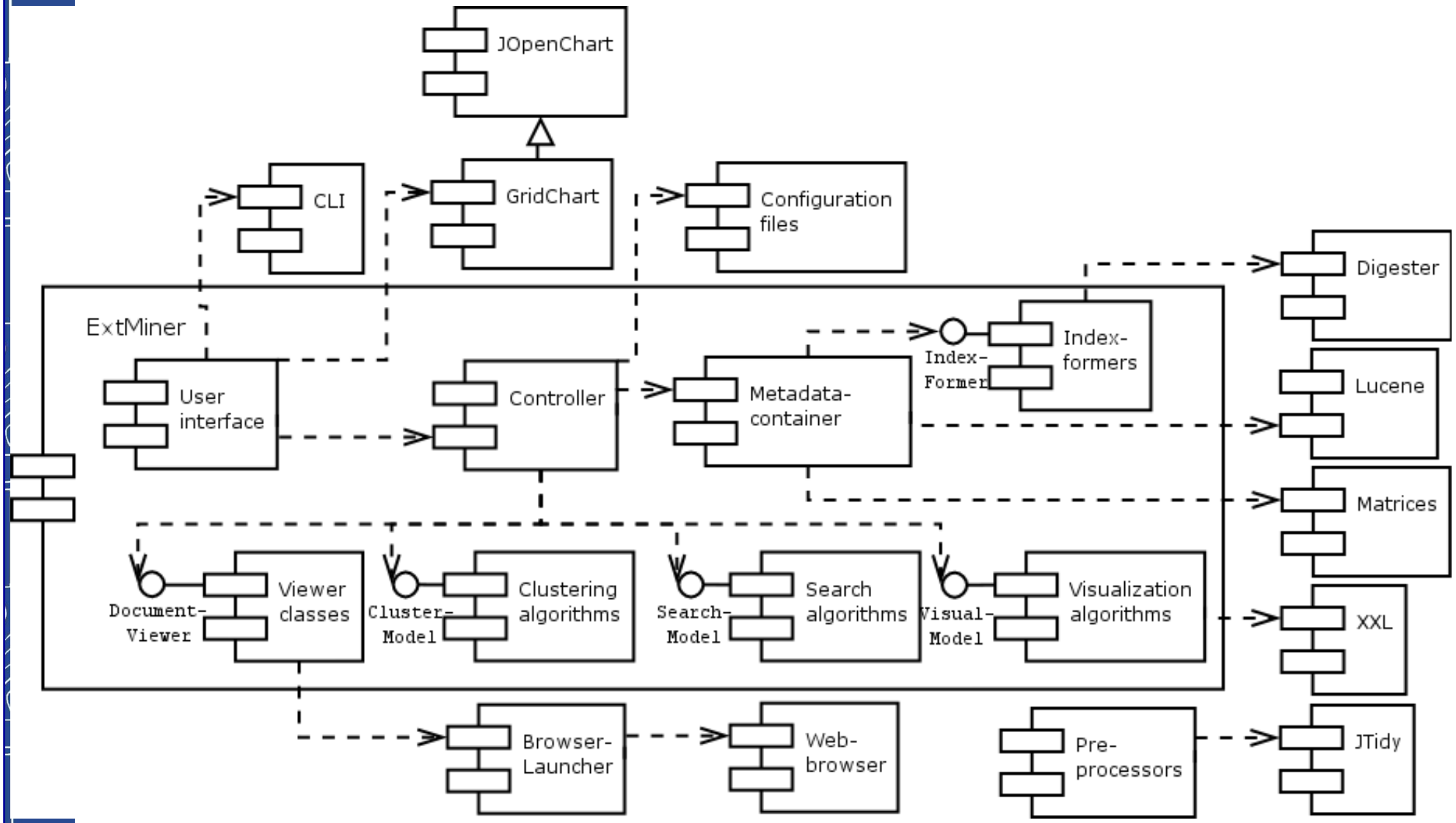
ExtMiner

- A platform and a proof-of-concept for combining
  - Different document features (eg. text, structure, links, metadata)
  - Ranking algorithms (eg. Cosine measure, PageRank)
  - Clustering algorithms (eg. DBSCAN, hierarchical clustering)
  - Visualization algorithms (eg. FastMap projection)
- Integrates many of the features previously implemented in separate systems
- Continuous search process based on ranked lists and cluster model



http://extminer.sf.net/

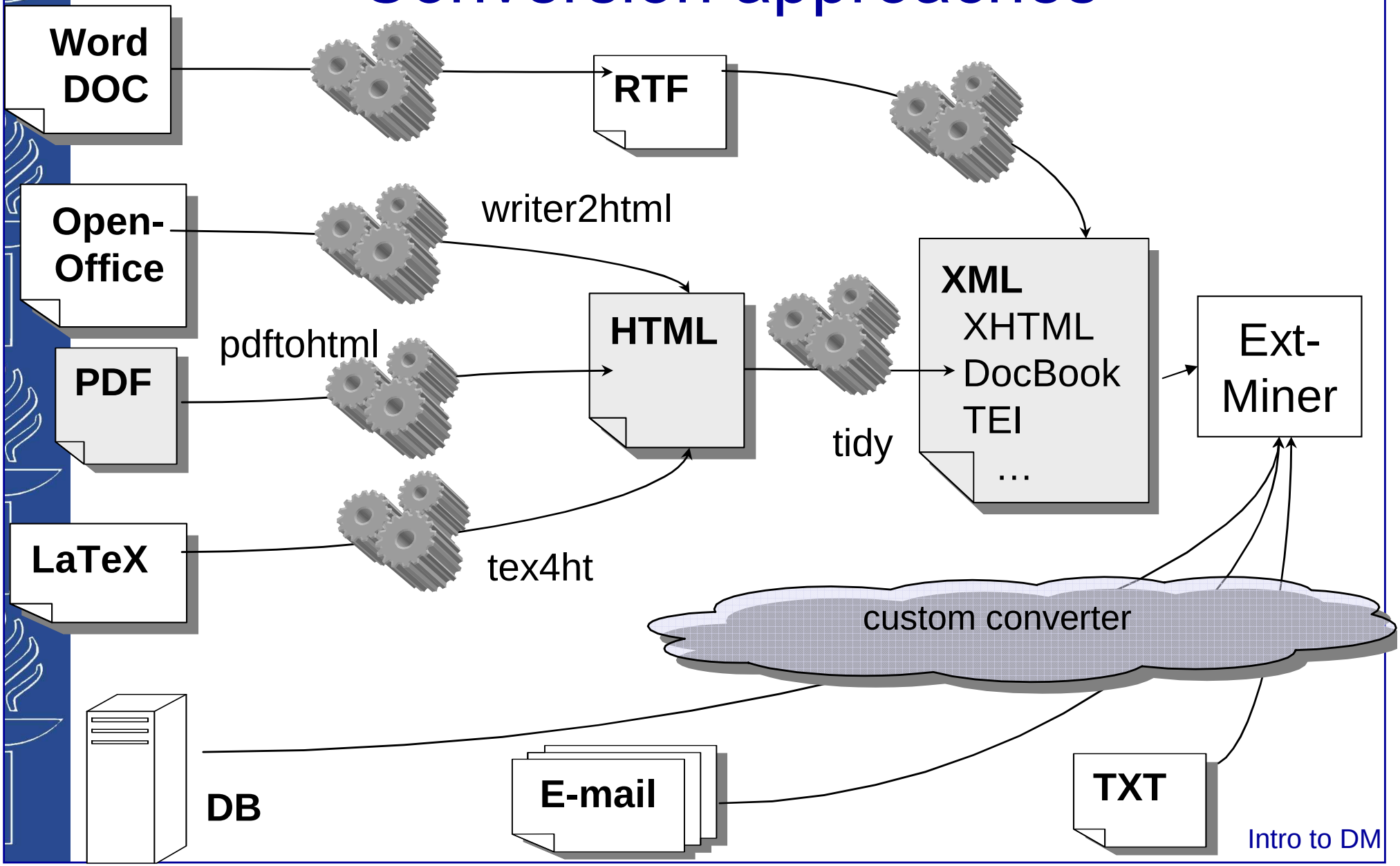
ExtMiner architecture



ExtMiner architecture (decomposed)

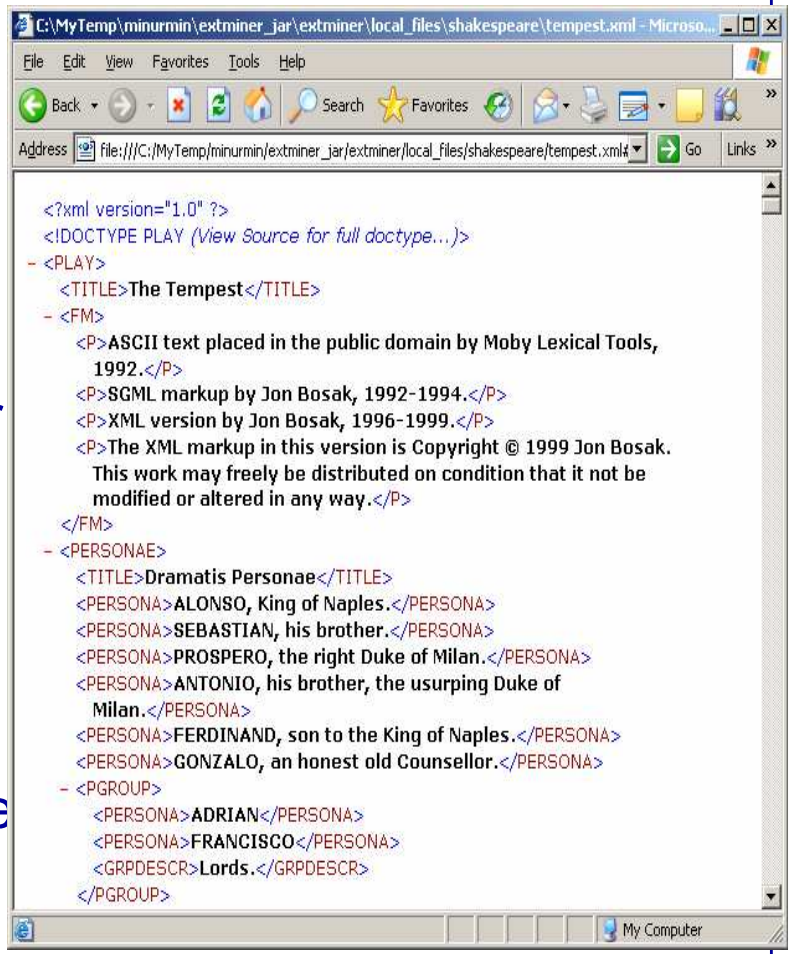
- 3 layers: UI, Application logic and Document index
- Document index consists of similarity matrices and a field-based term/link index
- Application logic includes pluggable ranking, clustering and visualization algorithms and extensible mechanism for index creation from various document repositories
- UI provides customizable views for documents, ranked search result list and cluster model tree
- Implemented with Java, published as open source. Third-party open source components (eg. Jakarta Lucene, JOpenChart) are utilized.

Conversion approaches



Indexing and configuration

- Documents must be available in a local filesystem
- Stemming,stopword removal and *tf*idf* weighting is performed by Lucene
- Digester handles rule-based XML parsing
- Documents are represented as field-based index (eg. tuples of vectors)
- Fields can be index terms, links, headers or document type –specific external metadata or structural information encoded as vectors
- Document-to-document similarities are precalculated for clustering
- Different index formers and field definitions can be utilized, depending on document type and application domain



Searching and clustering

- Extended vector model is applied both in ranking and clustering similarity calculation
- Let d be a document and q a query, both represented as tuples of n vectors (fields). Relevance estimate R is calculated as

$$R(d,q) = \sum_{k=1}^{n} w_k sim_k(r_k(d), r_k(q))$$

  - r denotes the restriction that extracts k-th vector from the tuple, sim is the similarity measure (such as boolean matching, cosine measure or co-citation), w denotes a field-specific weight supplied by the user (or matched evenly by default)
  - Substitute q with another document and you have a document-to-document similarity measure for clustering
  - Any metric clustering algorithm can be used, provided that the implemantation is available

User interface and visualization

- Iterative search and clustering process
  - –Search and clustering can be performed iteratively and focused to an appropriate subset of the collection
- Interactive cluster model
  - –The user can select documents from any of the views provided by the application: ranked list, cluster tree or visual projection. Cluster tree is interactive: a cluster can be marked as noise or subclusters of a single cluster can be merged (useful with hierarchical clustering)
- Simultaneous views for lists and clusters
  - –Both views are needed since lists and clusters support different search objectives. Clusters are easy to understand and help to cope with ambiguous terms, although they do not improve search quality as such.
- Any MDS (multidimensional scaling) –style projection algorithm can be used for visualization (currently FastMap)
- Documents can be opened in web browser or custom viewer (eg. text editor, XML tree view)