Formal Concept Analysis
with
Galicia

Petko Valtchev

&

The Galicia team

http://www.iro.umontreal.ca/~galicia/

Petko.Valtchev@UMontreal.CA
Overview

- **Formal concept analysis (FCA):** “application of lattice theory to data analysis”
  - Theory:
    - Back to work by O. Öre and by G. Birkhoff in 40s,
    - M. Barbut & B. Monjardet, R. Wille, B. Ganter, V. Duquenne…
  - Practice:
    - *social sciences*: Duquenne, Wille,…
    - *information retrieval*: Godin, Carpineto and Romano,…
    - *software engineering*: Godin, Snelting,…
    - *data mining*: Missaoui & Godin, Lakhal,…
  - Now:
    - rapidly growing community: “FCA” + “lattices” - couple of $10^3$ hits with Google,
    - annual forums: 2 intl. conferences, 2+ workshops,
  - Missing: a widely-shared software platform for FCA (ToscanaJ, ConExp, Galicia)
Outline of the Talk

- FCA: Galois connections, closures, lattices, min. generators …
- Computational challenges
- Realization within Galicia + demo
Formal Contexts and Galois Connections

K = (O, A, I)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>b</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>d</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Galois connection

Y ⊆ f(X) iff X ⊆ g(Y)

closure operators

X'' = g ° f(X)
Y'' = f ° g(Y)

closed sets

{a, d}'' = {a, d}
{5, 6}'' = {5, 6}

f(X) = X' = \{y ∈ A | ∀ x ∈ X, (x, y) ∈ I\}
g(Y) = Y' = \{x ∈ O | ∀ y ∈ Y, (x, y) ∈ I\}

f = g^{-1}

g
Lattices of Formal Concepts (« de Galois »)

**Families of closed**

\[ \mathcal{C}_K^o = \{ X \mid X \subseteq O, X'' = X \} \]
\[ \mathcal{C}_K^a = \{ Y \mid Y \subseteq A, Y'' = Y \} \]

**lattice (anti-)isomorphism**

\[ \mathcal{L}_K^o = (\mathcal{C}_K^o, \subseteq) \cong \mathcal{L}_K^a = (\mathcal{C}_K^a, \supseteq) \]

with \( f \) and \( g \) as co-bijections

**formal concept** \((X, Y)\)

\( X \in \mathcal{C}_K^o \) (extent), \( X = Y' \);
\( Y \in \mathcal{C}_K^a \) (intent), \( Y = X' \).

**partial order**

(sub-concept of)

\[(X_1, Y_1) \leq (X_2, Y_2) \iff X_1 \subseteq X_2 \]
\[
(\leftrightarrow Y_2 \subseteq Y_1)
\]

**lattice operators**

[Wille 82], [Barbut & Montjardet 70]

\[ \text{inf} - \bigcup_{j \in J}(X_j, Y_j) = (\bigcap_{j \in J}X_j, (\bigcup_{j \in J}Y_j)'' \]
\[ \text{sup} - \bigcup_{j \in J}(X_j, Y_j) = ((\bigcup_{j \in J}X_j)'', \bigcap_{j \in J}Y_j) \]
Equivalence Relation on $2^A$ Induced by $C^a_K$

Boolean lattice $2^A$

$\emptyset$

$A$

Closed sets:

bcd

Link crossing class border:

bcdg

bcdh

bcgh

bdgh

cdgh

bcd

bcg

bdg

bch

bdh

bgh

cdg

cdh

cgh

dgh

c

g

h

b

**Def.** A minimal generator $Z$ for a closed set $Y \subseteq A$ is a *minimal* subset of $Y$ such that $Z'' = Y$.

**Def.** $\text{Gen}_K$ = the family of minimal generators of all concept intents from $K$. 

---

**Minimal Generators**

- $\text{bcd}$
- $\text{bg}$
- $\text{bcd}$ (closed)
- $\text{bg}$ (min. generator)
- $\text{link crossing class border}$

---

**Diagram:**

- $\text{bcd}$
- $\text{bcg}$
- $\text{bdg}$
- $\text{bch}$
- $\text{bdh}$
- $\text{bgh}$
- $\text{cdg}$
- $\text{cdh}$
- $\text{cgh}$
- $\text{dgh}$

---

- **bcd**
- **bg**
- **closed**
- **min. generator**
- **link crossing class border**

---

- **bcdgh**
- **bcdh**
- **bcgh**
- **bdgh**
- **cdgh**
- **bcd**
- **bg**
- **bh**
- **cg**
- **ch**
- **g**
- **h**
- **c**
Why Are Min. Generators Interesting?

Minimal generators in...

• ...theory:
  • related to *minimal transversals* in hypergraph theory [Berge 89]
  • candidate keys of the tables in a *relational database*

• ... practice:
  • minimal sets of tests/exams/questions for a *medical diagnosis*

• ...algorithmic design:
  • *canonical representatives* for concept *intents*:
    • minimal generating *prefixes* in NextClosure [Ganter 84]
  • “*seeds*” for the computation of *intents*:
    • in general-purpose FCA algorithms: *Titanic* [Stumme et al 02]
    • in FCA-flavored *data mining* algorithms: *Close, Aclose* [Pasquier 00]
Implications

Given $K = (O,A,I)$, $Y, Z \subseteq A$, $Y \rightarrow Z$ is an implication:
- $Y$ premise,
- $Z$ conclusion.
(aka functional dependency in DB)

$\Sigma_K$ is large and redundant!

**Def.** $Y \rightarrow Z$ *valid* in $K$ if
\[ \forall o \in O, Y \subseteq o' \text{ forces } Z \subseteq o' \text{ (iff } Z \subseteq Y'). \]
$\Sigma_K$ = all *valid* implications of $K$.

**Ex.** bd $\rightarrow$ af, ae $\rightarrow$ cd: *valid*,
bc $\rightarrow$ agh: *invalid* (6 - ctr-ex.).

**Def.** A maximally informative rule:
- minimal premise,
- maximal conclusion.

**Ex.** bd $\rightarrow$ af: informative
ae $\rightarrow$ cd: not (e $\rightarrow$ acd *valid*).
Inference Axioms and Covers

**Def. Armstrong's axioms** for entailment

\[ \models \subseteq 2^{\Sigma_K} \times 2^{\Sigma_K} \]

Inference model (calculi) over \( \Sigma_K \)

- \( \emptyset \models Y \rightarrow Y \)
- \( Y \rightarrow Z, U \rightarrow V \models Y \cup U \rightarrow Z \cup V \)
- \( Y \rightarrow Z, U \rightarrow V, U \subseteq Z \models Y \rightarrow V \)

**Ex.**

\[ \text{bd} \rightarrow \text{af}, \ e \rightarrow \text{acd} \models \text{bde} \rightarrow \text{acdf} \]

**Def. Cover** for a set of implications

For \( \mathcal{I}, \mathcal{J} \subseteq \Sigma_K \), \( \mathcal{I} \) is a **cover** of \( \mathcal{J} \) iff \( \mathcal{I} \models \mathcal{J} \)

OEWG'05, DIMACS, March 2005
Pseudo-closed Sets and Canonical Basis

**Def.** \(\hat{\Phi}_K \subseteq 2^A\): the pseudo-closed sets of \(K\):
- \(Y \neq Y''\),
- for all \(Z\) pseudo-closed, \(Z \subset Y\) forces \(Z'' \subset Y\).

**Def.** (Duquenne & Guigues 86)

**Canonical basis** of \(K\), \(\hat{\Phi}_K = \{Z \rightarrow Z'' \mid Z \in \hat{\Phi}_K\}\).

**Prop.** For all \(K\), \(\hat{\Phi}_K\) is a cover of \(\Sigma_K\) of a minimal size (nb. of rules).

**Ex.** The basis of the example

\[
\begin{align*}
\text{adg} & \rightarrow \text{bcefhi} & \text{acq} & \rightarrow \text{h} & \text{ah} & \rightarrow \text{g} & \rightarrow \text{a} \\
\text{acdef} & \rightarrow \text{bghi} & \text{abd} & \rightarrow \text{f} & \text{ae} & \rightarrow \text{cd} & \text{af} & \rightarrow \text{d} \\
\text{abcghi} & \rightarrow \text{def} & \text{ai} & \rightarrow \text{cgh}
\end{align*}
\]

**Ex.** acdef in \(\hat{\Phi}_K\):

- \(ae, af\) in \(\hat{\Phi}_K\);
  - \(ae'' = \text{acde} \subset \text{acdef}\),
  - \(af'' = \text{afd} \subset \text{acdef}\).
**Partial Implications and Further Bases**

**Def. Partial implication** \( X \rightarrow Y \) (Luxenburger 92)  
Not valid to 100% (exists object \( o : X \subseteq o' \), but \( Y \not\subseteq o' \)).

Two bases for partial implications, following the **lattice structure** [Luxenburger 92]

**Def. Global basis** :
\[ \{ Z_1'' \rightarrow Z_2'' - Z_1'' | Z_1'' \subset Z_2'' \} \].

- \( bcd \rightarrow aefgh \)  
  \( Z = bcd, Y = abcdefgh \)

- \( bcd \rightarrow a \)  
  \( Z = bcd, Y = abcd \)

**Def. Cover basis** :
\[ \{ Z'' \rightarrow Y'' - Z'' | Z'' \text{ minimal closed subset of } Y'' \} \].

a.k.a **association rules**
Why Study the Pseudo-closed?

Pseudo-closed in...

• ...theory:
  • related to the precedence relation in the lattice of all closures on a ground set A [Caspard & Monjardet 03]
  • minimal covers for functional dependencies in relational databases [Maier 80]

• ...algorithmic design:
  • alternative closure computation mechanism for intents:
    • helps restrict usage of extents in large datasets [Valtchev & Duquenne 03],

• ... practice:
  • non-redundant sets of association rules in data mining [Kryszkiewicz 02]
Families not necessarily disjoint:
- Only \( \mathcal{C}_K \cap \not\mathcal{C}_K = \emptyset \)
- \( \mathcal{Gen}_K \) may share elements with both other families

**Prop.** \( \mathcal{Gen}_K \) is an order ideal of the Boolean lattice \( 2^A \):
\[ Z \in \mathcal{G}_K \text{ forces } \forall Y \subseteq Z, Z \in \mathcal{Gen}_K. \]

**Prop.** \( \not\mathcal{C}_K \cup \mathcal{C}_K \) is closed for intersection (closure space):
\[ \not\mathcal{C}_K \cup \mathcal{C}_K = (\not\mathcal{C}_K \cup \mathcal{C}_K) \cap. \]

**Prop.** Individual elements of \( \not\mathcal{C}_K \) preserve the closure property:
\[ \forall Y \in \not\mathcal{C}_K, \forall Z \in \mathcal{C}_K, Y \cap Z \in \mathcal{C}_K \cup \{Y\}. \]
Outline of the Talk

- FCA: Galois connections, closures, lattices, min. generators …
- Computational challenges
- Realization within Galicia
## Algorithmic Problems in FCA

<table>
<thead>
<tr>
<th>Mode</th>
<th>Concept set $C_K$</th>
<th>Concept set + precedence $L_K = (C_K, \leq)$</th>
<th>Min. generators $Gen_{L_K}$</th>
<th>Canonical basis $\hat{B}_K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>$NextClosure$</td>
<td>[Bordat 86], [Nourine &amp; Raynaud 99]</td>
<td>Titanic [Stumme et al 02], [Pfalz &amp; Taylor 02]</td>
<td>NextClosure for PC [Ganter 84]</td>
</tr>
<tr>
<td>on-line</td>
<td>$O$</td>
<td>[Godin et al 95], [Carpinetto &amp; Romano 96], [Valtchev et al 02, 03]</td>
<td>[Valtchev et al 04],</td>
<td></td>
</tr>
<tr>
<td>on-line</td>
<td>$A$</td>
<td>[Nehme et al 05]</td>
<td>[Nehme et al 05]</td>
<td>[Ob’edkov &amp; Duquenne 03]</td>
</tr>
<tr>
<td>on-line</td>
<td>$O$</td>
<td>[Valtchev &amp; Missaoui 01]</td>
<td>[Frambourg et al, submitted]</td>
<td></td>
</tr>
<tr>
<td>on-line</td>
<td>$A$</td>
<td>[Valtchev &amp; Duquenne 03]</td>
<td>[Valtchev et al 02]</td>
<td></td>
</tr>
</tbody>
</table>

OEWG’05, DIMACS, March 2005
NextClosure

- **Reference algorithm in FCA:** [Ganter 84]

- Typical combinatorial generation (listing) procedure:
  - Search for **closed attribute sets** throughout the Boolean lattice \(2^A\),
  - Attribute set \(A\) **totally ordered**,  
  - **Closures** of candidate sets computed,  
  - Closed sets listed in a **lexicographic** order:  
    - Implicit **tree structure**  
  - Looking for a **canonical representative** for each closed set:  
    - a minimal generating prefix = minimal prefix including a **minimal generator**  
    - pruning the search tree  
  - Uses **no memory**:  
    - moves from one candidate to the next one in the **lexicographic** order,  
    - hence suitable for **large lattices**,
On-line Maintenance of Lattices & Co.

Why?

- **Natural evolution** in a dataset:
  - organizations feed new data to their databases on a regular basis,
  - reuse of current analysis results instead of computing the new ones from scratch,

- **Explorative** analysis:
  - adding/removing input data elements,
  - tracking the changes in the result,

- **Potential efficiency** gains:
  - *Incremental* mode: much faster than batch reconstruction from scratch,
  - *Batch* mode: provably faster for sparse data,
On-line Lattice Maintenance

$K_1 = (O, A, I)$

$K_2 = (O, A \cup \{a\}, I \cup a \times a')$

**Problem:** Given $\mathcal{L}_1$ and $(a, a')$, transform the data structure representing $\mathcal{L}_1$ into an equivalent for $\mathcal{L}_2$. 
The Approach Foundations

Idea: Object dimension stable. Work amounts to add a new extent to $C_1^o$ and close the result by $\cap$:

$$C_2^o = (C_1^o \cup a^*)^\cap$$

New extents: $C_2^o - C_1^o$

$\Rightarrow$ new concepts: $N_2(a)$

Existing extents: $C_2^o \cap C_1^o$

<table>
<thead>
<tr>
<th>Transition</th>
<th>$L_1 \rightarrow L_2$</th>
<th>Old</th>
<th>Genitor</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>intent</td>
<td>same</td>
<td>same</td>
<td>change</td>
<td></td>
</tr>
<tr>
<td>extent</td>
<td>same</td>
<td>same</td>
<td>same</td>
<td></td>
</tr>
<tr>
<td>lower cov.</td>
<td>same</td>
<td>change</td>
<td>same</td>
<td></td>
</tr>
<tr>
<td>upper cov.</td>
<td>same</td>
<td>change</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notation</td>
<td>$U_2(a)$</td>
<td>$G_2(a)$</td>
<td>$M_2(a)$</td>
<td></td>
</tr>
</tbody>
</table>

OEWG'05, DIMACS, March 2005
The Approach Foundations (cont’d)

Idea: Find the homologous concepts of genitors and modified in $\mathcal{L}_2$ and carry out the restructuring from them on, up to obtaining $\mathcal{L}_3$.

Equivalence relation on $\mathcal{L}_1$, by extent intersection with $a'$:

$$[c]_a = \{c \in C_1 \mid \text{ext}(c) \cap a' = \text{ext}(c) \cap a'\}$$

Characterization of $G_1(a)$ and $M_1(a)$:
Minima in their equivalence classes $[\_]_a$

$$c \in G_1(a) \cup M_1(a) \iff c = \text{min}([c]_a)$$
Lattice Update Method: Attribute-wise

Procedure Add-Attribute(
  \textbf{Input:} \mathcal{L} a lattice, \( a \) an attribute;
  \textbf{Output:} \mathcal{L} a lattice, \textit{updated})

for each \( c = (X, Y) \) in \( \mathcal{L} \)
  
  \( E \leftarrow X \cap a' \)
  
  if \( c \) \textit{minimal} for \( E \) then
    if \( X = E \) then \quad // \textit{modified}
      \textbf{Update}(c)
    else \quad // \textit{genitor}
      \( cc \leftarrow \text{new-concept}(E, Y \cup \{o\}) \)
      \( \mathcal{L} \leftarrow \mathcal{L} \cup \{cc\} \)
      \textbf{UpdateOrder}(c, cc)

**Problem_1:** Fit min. generator \( \text{Gen}_{\mathcal{L}} \)
computation to Add-Attribute(\( \mathcal{L}, a \)).

See [Nehme et al. 05]

**Problem_2:** Fit pseudo-closed \( \text{PC}_{\mathcal{L}} \)
computation to Add-Attribute(\( \mathcal{L}, a \)).

See [Ob’edkov & Duquenne 03]
Merge of Lattices & Co.

Why?

◆ looking for the **interactions among subsets of descriptors** in a dataset:
   - *split* the descriptor set,
   - *process* the resulting subsets:
     » first independently (**factor** lattices),
     » then as a whole (**global** lattice),
   - *map* the **factor** lattices into the **global** one,
   - *merge-based* construction = last two steps carried out **simultaneously**.

◆ **visualization** (related to previous topic):
   - present the global lattice as "projected" into the **direct product** of the factors,

◆ **potential efficiency gains**: take advantage of distributed/parallel architecture
   - split the work into sub-problems,
   - deal with them separately,
   - put together the partial results,
Fragmentation of Contexts

**Apposition** = recompose a context after a *split*

\[ K = K_1 \mid K_2 \]

**K** = (O, A, I)

\[ K_1 = (O, A_1, I \cap O \times A_1) \]

\[ K_2 = (O, A_2, I \cap O \times A_2) \]
Lattice Merge
The Problem

Notations:
- **Contexts**: factors $K_1$, $K_2$, *global* $K_3 = K_1 | K_2$.
- **Closures**: operators $\_i^i$ (i=1,2,3).
- **Lattices, canonical bases, generators**: 
  - factors $\mathcal{L}_i / \mathcal{F}_i / \mathcal{Gen}_1$ (i=1,2),
  - *global* $\mathcal{L}_3 / \mathcal{F}_3 / \mathcal{Gen}_3$,
  - *direct product* $\mathcal{L}_{1,2} / \mathcal{F}_{1,2}$.

Given:
- **Factor lattices**: $\mathcal{L}_1$, $\mathcal{L}_2$
- *(OPT) canonical bases of factors*: $\mathcal{F}_1$, $\mathcal{F}_2$
- *(OPT) min. generator families of factors*: $\mathcal{Gen}_1$, $\mathcal{Gen}_2$

Find:
- **Global lattice**: $\mathcal{L}_3$
- *(OPT) global canonical base*: $\mathcal{F}_3$
- *(OPT) global min. generator family*: $\mathcal{Gen}_3$
Prop. \( \mathcal{L}_3 \) is a sub-semi-lattice of \( \mathcal{L}_{1,2} \) hence may be embedded into it.
Approaching the Merge

Complete lattice merge, i.e., concepts and order

Key ideas:

- **Mixture of extent families**: $C^o_3 = \text{all pair-wise intersections on } C^o_1 \times C^o_2$.
- **Each global extent (3-extent) $Y$**: generated by a set of pairs.
- **Canonical element** of $C^o_1 \times C^o_2$:
  - the minimum of all pairs $(\hat{Y}_1, \hat{Y}_2)$ from $C^o_1 \times C^o_2$ generating a 3-extent $Y$.
- Completing the concept $(Y, Y^3)$: the **intent** $Y^3$ is the union of canonical intents:
  - $Y^3 = \hat{Y}^1_1 \cup \hat{Y}^2_2$.
Merge: 3-step Construction Procedure

1. Identify concepts
2. Compute intents & extents
3. Detect precedence links
Outline of the Talk

- **FCA:** Galois connections, closures, lattices, min. generators …

- **Computational challenges**

- **Realization within Galicia**
Goals of the Galicia project

Develop a tool set to support:

- **Research** on FCA theory and algorithms for the analysis of:
  - **structured** data formats (*data and meta-data*):
    » relational DB, UML models, image meta-data, etc.
  - **semi-structured** data formats (*data and meta-data*):
    » OWL, RDF(S), XMI, etc.
  - volatile datasets,
  - large databases,

- **Practical applications** of FCA techniques to:
  - Data analysis and mining in:
    » Software engineering,
    » Bioinformatics,
    » Image retrieval and mining,
    » Ontology construction.
Member Teams

- Université de Montréal (Qc, CA)
  - P. Valtchev (Assist. Prof.),

- Université du Québec à Montréal (CA)
  - R. Godin (Prof.)

- Université du Québec en Outaouais (CA)
  - R. Missaoui (Prof.)

- LIRMM, Montpellier (FR)
  - M. Huchard (Prof.)

- Université de la Réunion (FR)
  - D. Grosser (Assist. Prof.)

- LORIA, Nancy (FR)
  - A. Napoli (Sen. Res.)
Life-cycle of a Lattice/Rule Set

1. Prepare data
2. Construct concept hierarchy/rule set
3. Visualize results
The Galicia Platform

Rich set of tools for lattices, semi-lattices, general posets, rule bases, etc. :

- **Open-source**
  
  [http://www.iro.umontreal.ca/~galicia](http://www.iro.umontreal.ca/~galicia) (Home Page of the platform)
  
  [https://sourceforge.net/projects/galicia/](https://sourceforge.net/projects/galicia/) (Home Page of the SF project)

- **Portable**: developed in Java,

- **Generic**: abstract types, implementations easily exchangeable.

- Supports different input data formats:
  - Binary data
  - Categorical data
  - Relational Context Families: entities + relations
Key Functions of Galicia

**Context** import/export and edition:
- binary,
- *relational* and *multi-valued*

Construction of lattices and derived structures:
- **Lattice** construction:
  - *Batch* mode
  - *Incremental*: object- and attribute-wise
  - *Merge-based*: object- and attribute-wise
- Galois sub-hierarchies
- **Iceberg** lattices

Association rule extraction from the lattice of intents:
- **Exact rules (valid implications)**: *Duquenne-Guigues* basis [Guidues & Duquenne 86], *generic* basis [Pasquier et al. 99].
- **Approximate rules (partial implications)**: *Luxenburger* bases [Luxenburger 92], *informative* basis [Pasquier et al. 99].
Exploration of FCA Results

Structure visualization and navigation services:

- **Diagram types:**
  - Standard Hasse diagrams,
  - *Nested Line Diagrams (work in progress)*,

- **Layout mechanisms** for layered diagrams:
  - Static/dynamic formatting,
  - Layered,
  - Magnetism (attraction - repulsion model).

- **Views:** 2D, 3D, 3D + rotation.

- **Navigation:**
  - hierarchy overview.

**I/O operations** for various formats:
- dedicated data formats: SLF (*in-house*), IBM,
- XML-based formats: XML DTDs for input data and posets, RCF (*in-house*).
Demo of Galícia

Galois Lattice Interactive Constructor
On-going research projects:

- **Relational** FCA: bring FCA and conceptual data models (UML, E-R, etc.) closer:
  - *Recursive* and *circular* links in data,
  - *Co-definition* of concepts on different sorts of objects:
    » Ex. Customer, Transaction, Product,
  - Iterative (fixed-point) construction of a set of related lattices
  - ... and a bunch of unresolved problems...

- **Evolution** of association rule bases:
  - Merge of *factor bases* along:
    » the *object* dimension,
    » the attribute dimension,
  - Decomposition of lattices/posets along different operators
On-going application projects involving *Galicia*:

- **Re-engineering** of software *analysis-level models*: extracting high level abstractions from existing conceptual models described in UML;

- **Image retrieval and mining**: lattice products to detect and visualize interactions between lower level and higher level image characteristics,

- **Information (text) retrieval**: query analysis and expansion

- **Bio-informatics**: mining 3D structure of proteins (*initial stage*),