Towards a flexible author name disambiguation framework

Łukasz Bolikowski and Piotr Jan Dendek

Interdisciplinary Centre for Mathematical and Computational Modelling,
University of Warsaw

DML 2011: Towards a Digital Mathematics Library,
20-21 July 2011

Motivation

- **Information storage shortcomings**
  1. Authors unique id
  2. Binding between author and his articles
  3. Authors' id mapping across libraries
- **Problem with name representation**
  1. Different forms
     - "M. Brown"
     - "Michael Brown"
     - "M. A. Brown"
  2. metadata extraction deficiencies
     - OCR
     - defective handling of diacritics
- **Solution**
  - Name Disambiguation

Problem statement

- Three stages of creating the solution
  1. Affinity measures definition
  2. Training
  3. Infrastructure

Definitions

- **Author.** A human who writes an article/articles.
- **Contribution.**
  - An author's signature on his paper.
  - A unique contribution id: document id + author's order on the list of authors
  - Example: an article with **id 123** has two authors **R. Black** and **J. Smith**.
  - Contributions: **123#1** and **123#2**
- **Shard.** A group of contributions which shares the same hash function result (e.g. having same surname).
- **Identity.** A group of contributions (stored in the same shard!) believed to be done by the same person.

Definitions

- **Author.** A human who writes an article/articles.
- **Contribution.**
  - An author's signature on his paper.
  - A unique contribution id: document id + author's order on the list of authors
  - Example: an article with **id 123** has two authors **R. Black** and **J. Smith**.
  - Contributions: **123#1** and **123#2**
- **Shard.** A group of contributions which shares the same hash function result (e.g. having same surname).
- **Identity.** A group of contributions (stored in the same shard!) believed to be done by the same person.

Definitions

- **Author.** A human who writes an article/articles.
- **Contribution.**
  - An author's signature on his paper.
  - A unique contribution id: document id + author's order on the list of authors
  - Example: an article with **id 123** has two authors **R. Black** and **J. Smith**.
  - Contributions: **123#1** and **123#2**
- **Shard.** A group of contributions which shares the same hash function result (e.g. having same surname).
- **Identity.** A group of contributions (stored in the same shard!) believed to be done by the same person.

Definitions

- **Author.** A human who writes an article/articles.
- **Contribution.**
  - An author's signature on his paper.
  - A unique contribution id: document id + author's order on the list of authors
  - Example: an article with **id 123** has two authors **R. Black** and **J. Smith**.
  - Contributions: **123#1** and **123#2**
- **Shard.** A group of contributions which shares the same hash function result (e.g. having same surname).
- **Identity.** A group of contributions (stored in the same shard!) believed to be done by the same person.

Definitions

- **Author.** A human who writes an article/articles.
- **Contribution.**
  - An author's signature on his paper.
  - A unique contribution id: document id + author's order on the list of authors
  - Example: an article with **id 123** has two authors **R. Black** and **J. Smith**.
  - Contributions: **123#1** and **123#2**
- **Shard.** A group of contributions which shares the same hash function result (e.g. having same surname).
- **Identity.** A group of contributions (stored in the same shard!) believed to be done by the same person.

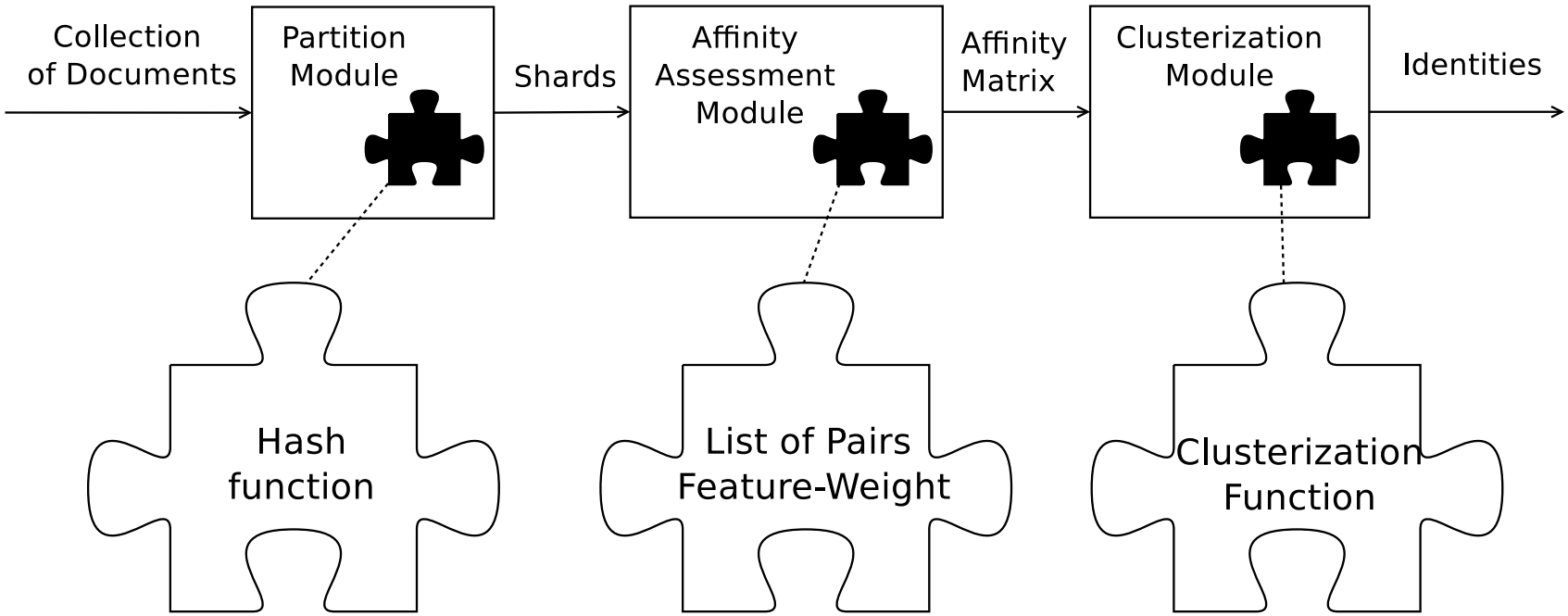
Definitions

- **Author.** A human who writes an article/articles.
- **Contribution.**
  - An author's signature on his paper.
  - A unique contribution id: document id + author's order on the list of authors
  - Example: an article with **id 123** has two authors **R. Black** and **J. Smith**.
  - Contributions: **123#1** and **123#2**
- **Shard.** A group of contributions which shares the same hash function result (e.g. having same surname).
- **Identity.** A group of contributions (stored in the same shard!) believed to be done by the same person.

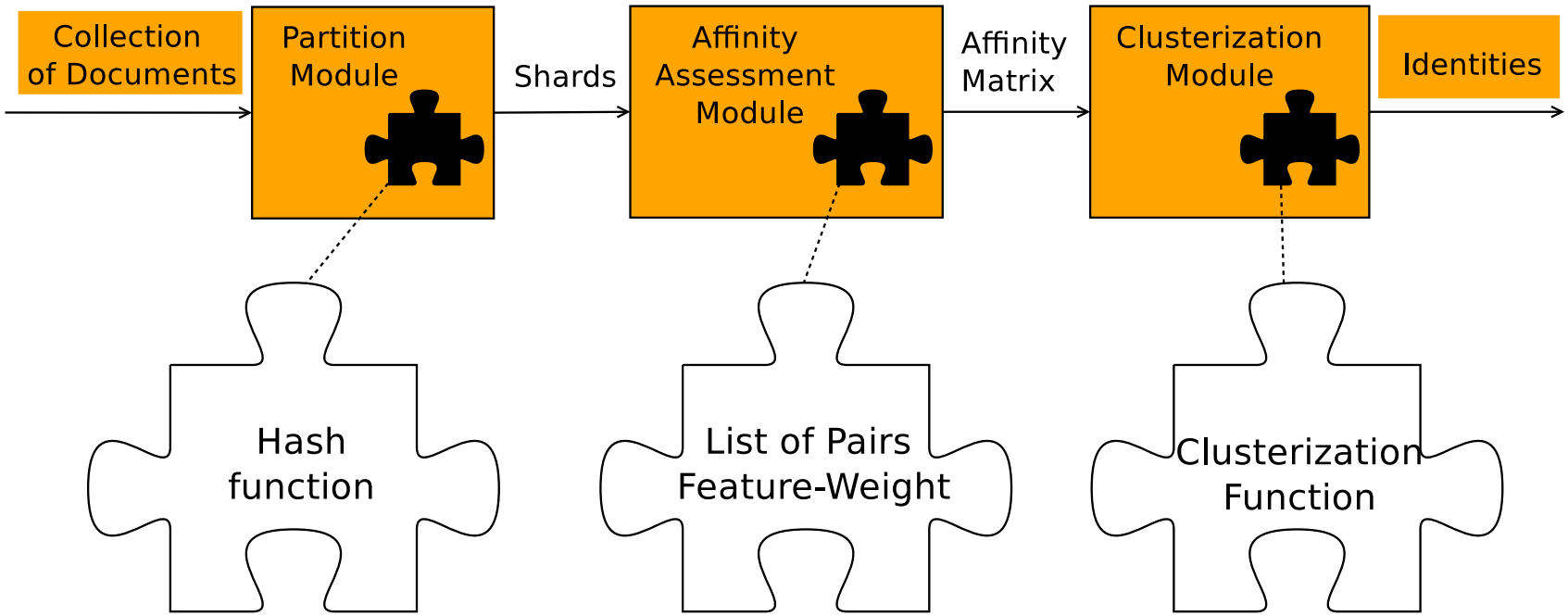
Definitions (continued)

- **Feature.**
  - A function which compares two contributions with respect to some field(s) (e.g. year of publication)
  - Result: a value between $[-1, 1]$
    - -1 contributions made by different authors
    - 0 undetermined
    - 1 contributions are made for sure by same author
- **Weight.**
  - An indicator of feature's importance.
  - A non-negative real number.
- **Atomic affinity.** A product of feature's result and weight.
- **Total affinity.** The sum of atomic affinities.

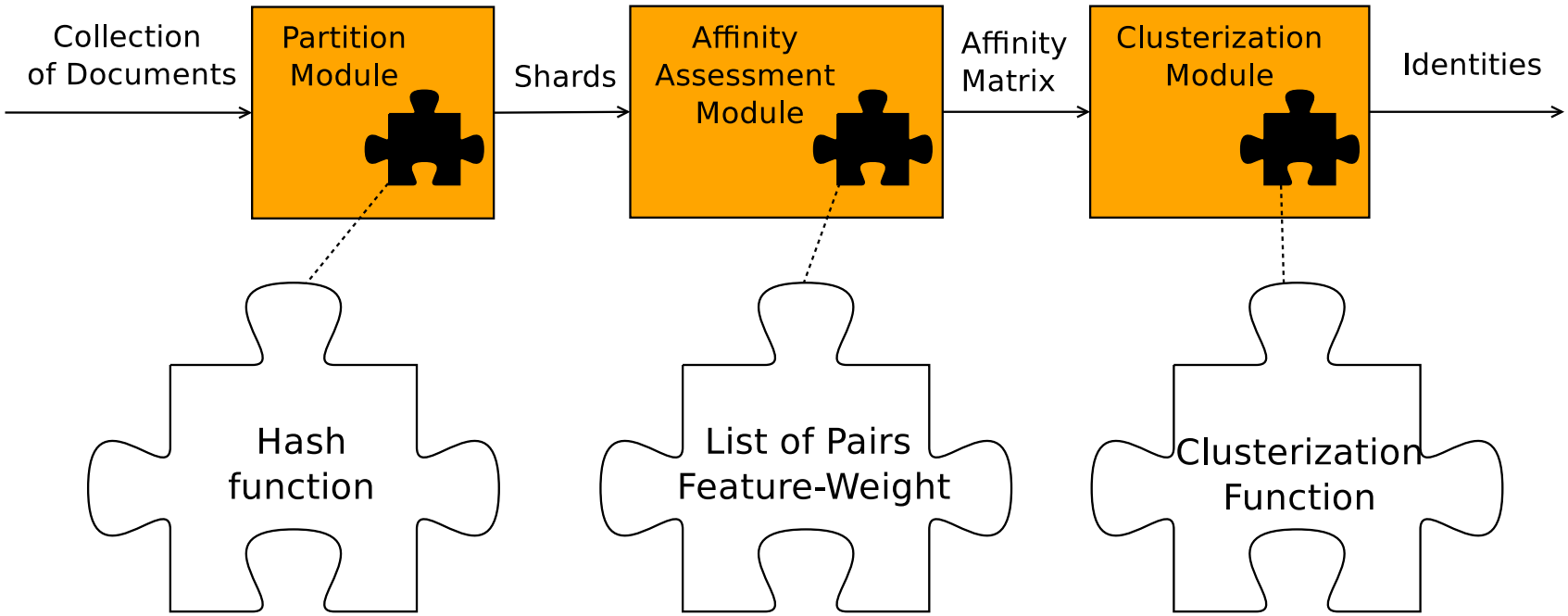
Process stages



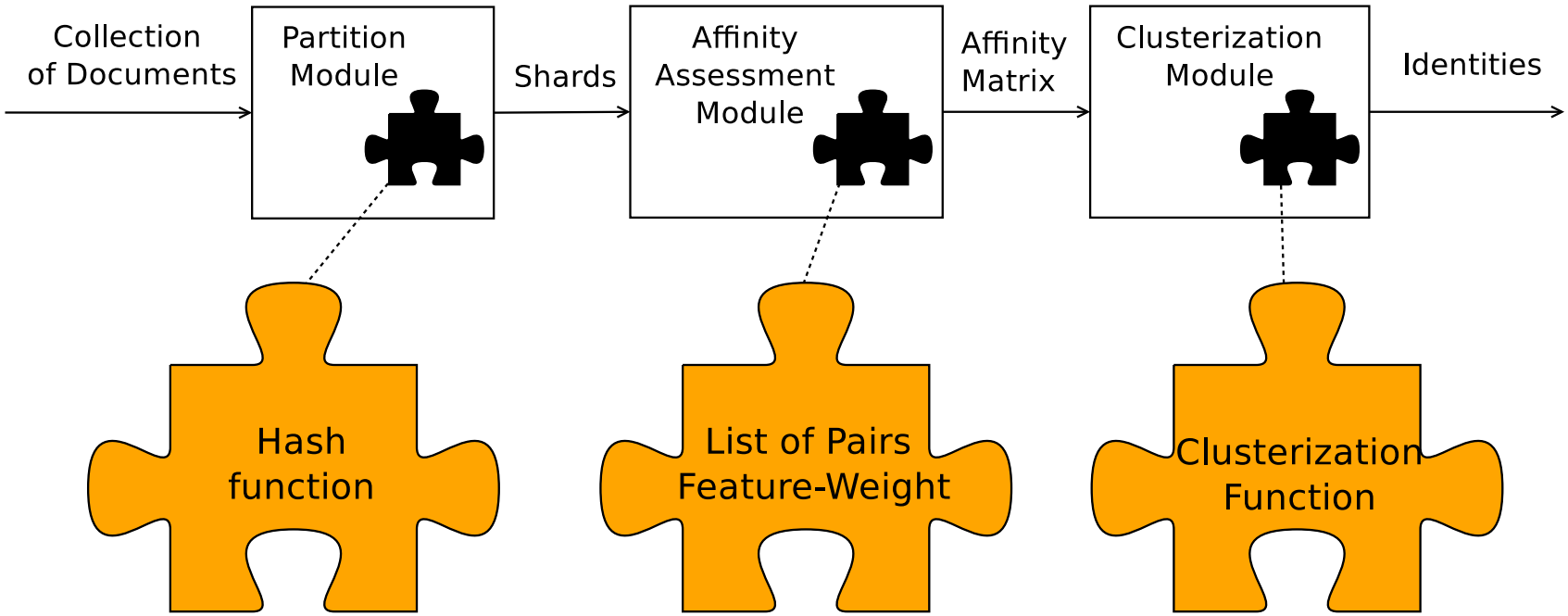
Process stages



Process stages

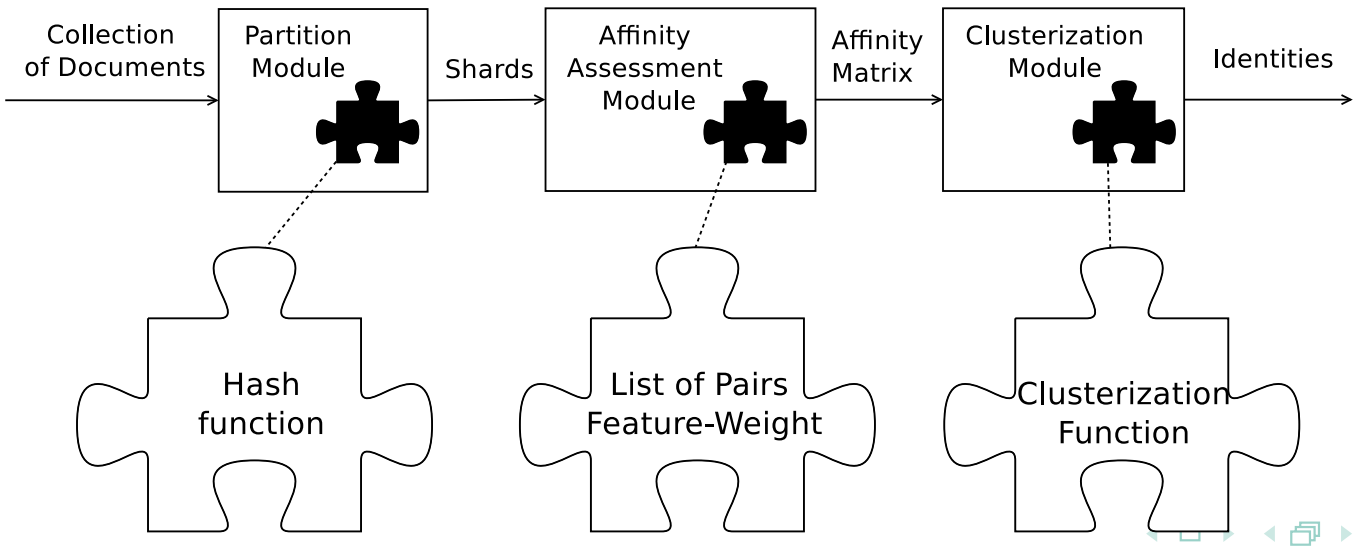


Process stages



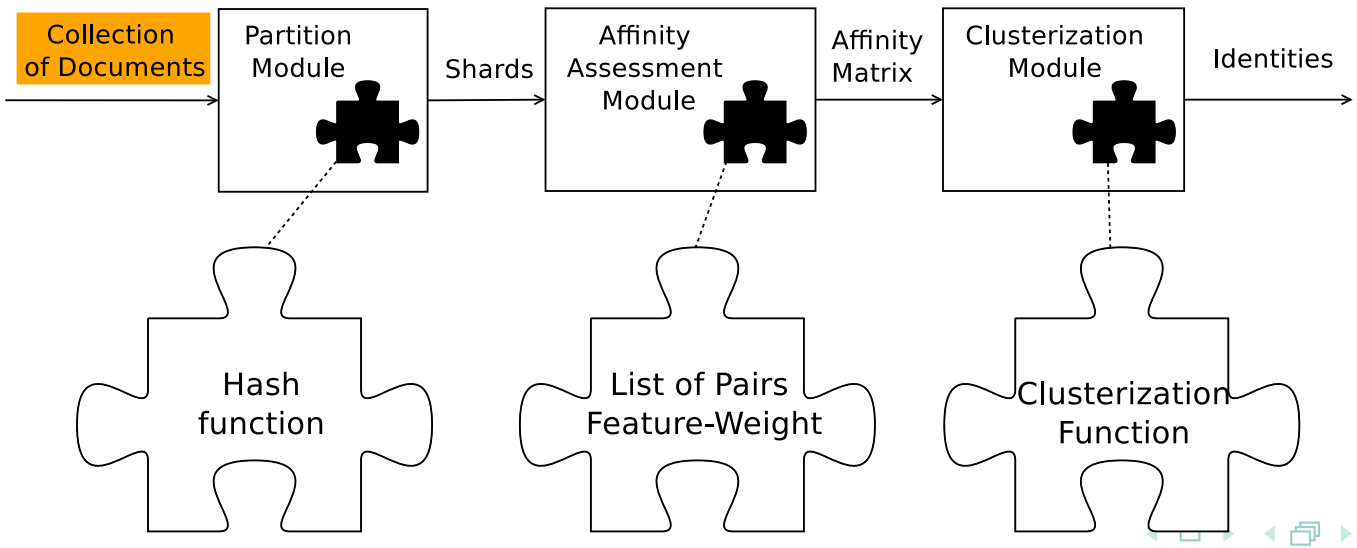
Process stages

1. Data import
2. Contributions decomposition
3. Affinity calculation
4. Clusterization
5. Result persistence



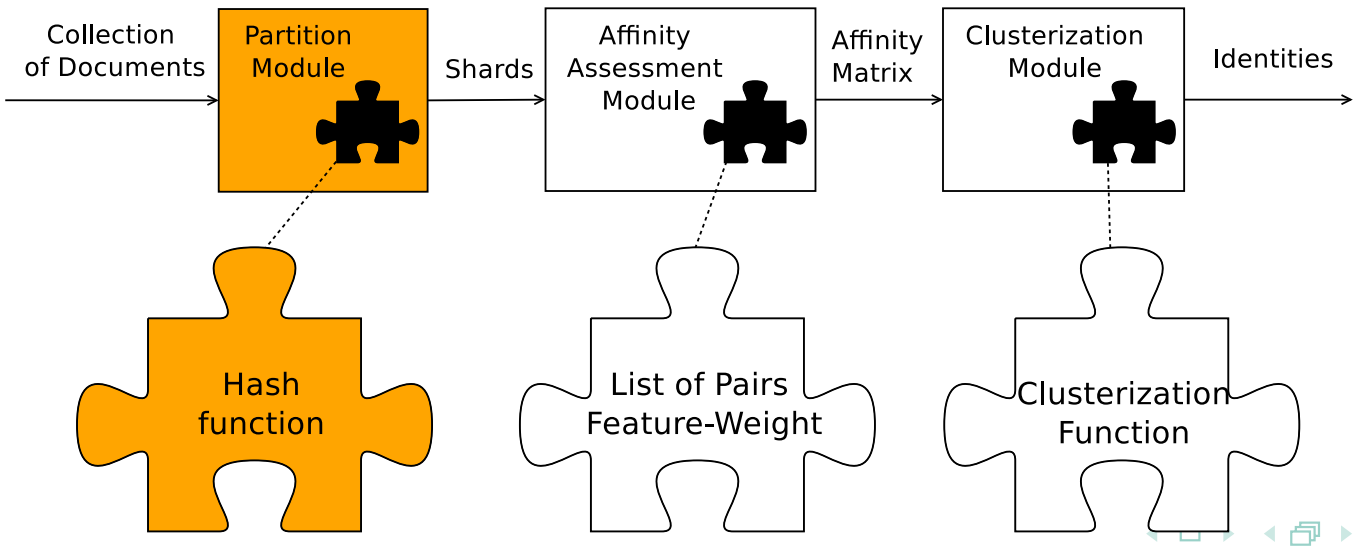
Process stages

1. Data import
2. Contributions decomposition
3. Affinity calculation
4. Clusterization
5. Result persistence



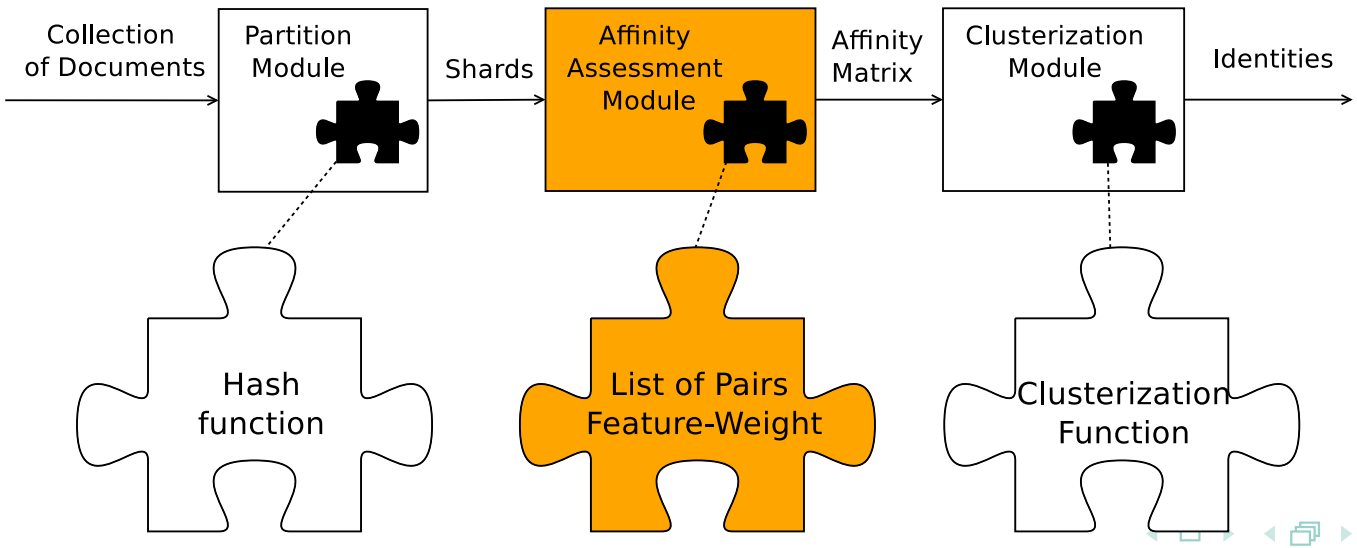
Process stages

1. Data import
2. Contributions decomposition
3. Affinity calculation
4. Clusterization
5. Result persistence



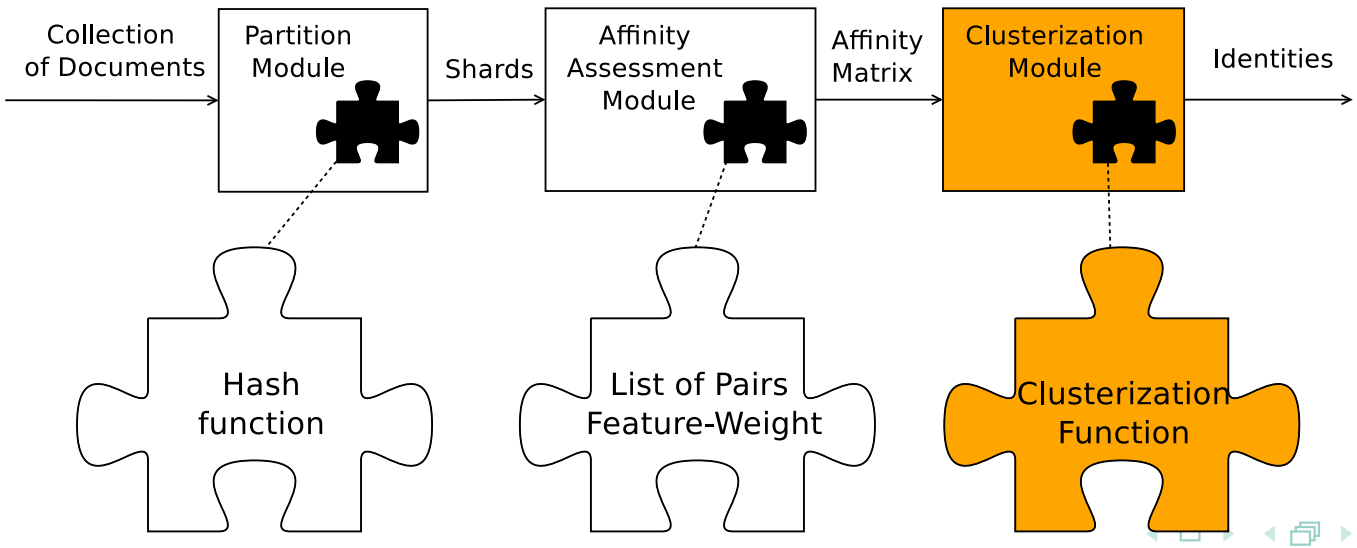
Process stages

1. Data import
2. Contributions decomposition
3. Affinity calculation
4. Clusterization
5. Result persistence



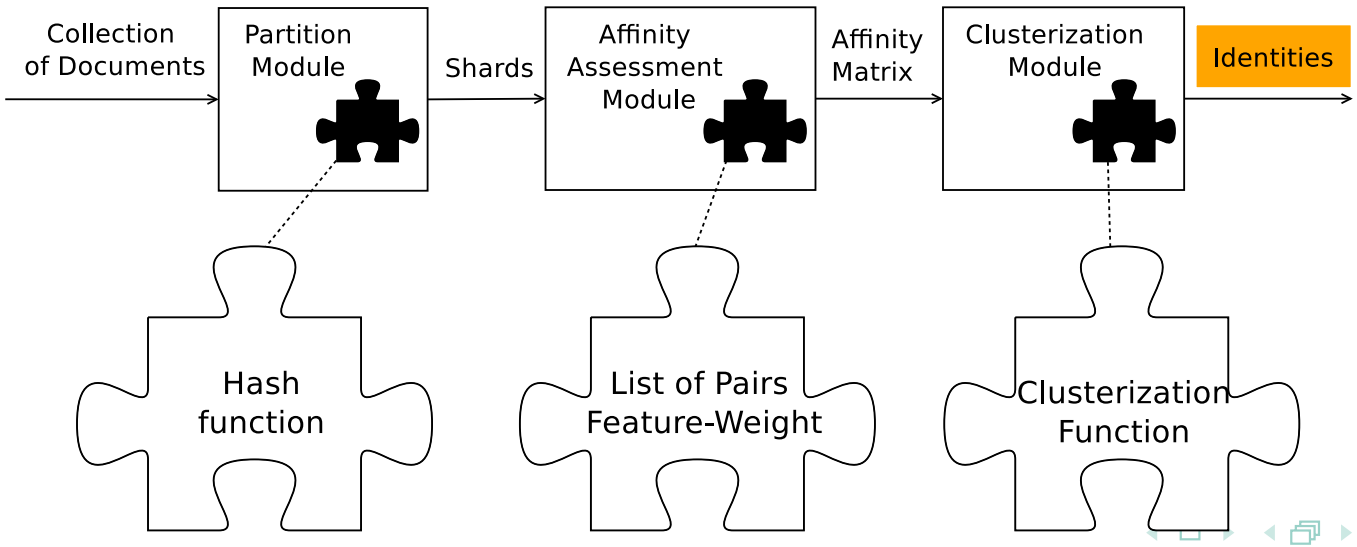
Process stages

1. Data import
2. Contributions decomposition
3. Affinity calculation
4. Clusterization
5. Result persistence

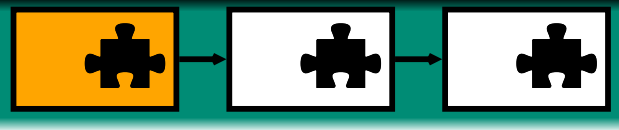


Process stages

1. Data import
2. Contributions decomposition
3. Affinity calculation
4. Clusterization
5. Result persistence

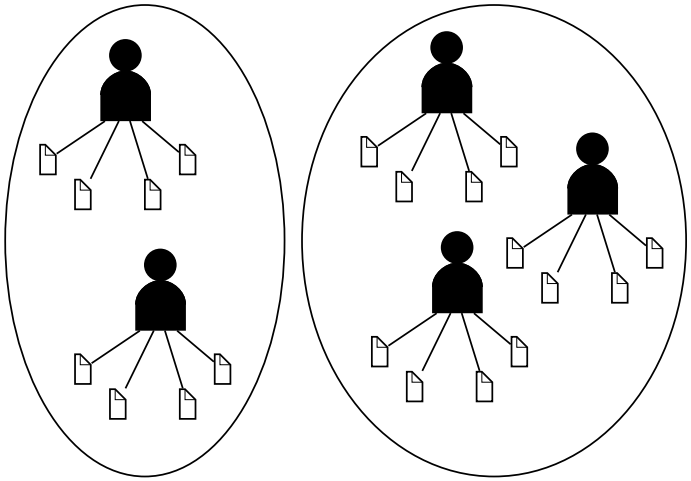


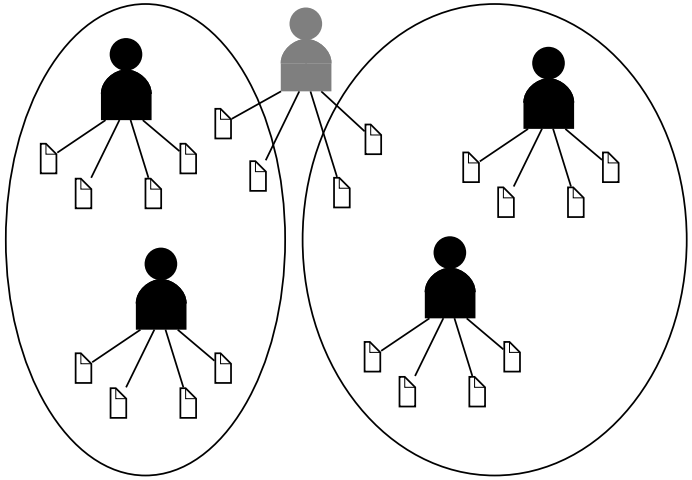
Hash function



- **Goal** - ability to cope with a large dataset
- **Requirements**
  - Contributions of the same person must be in the same subset (shard)
  - A shard may contain contributions of more then one author
  - Shards should not overlap!
- A hash function **example**:
  - function which returns the lower-cased surname of a name with all the diacritic marks removed

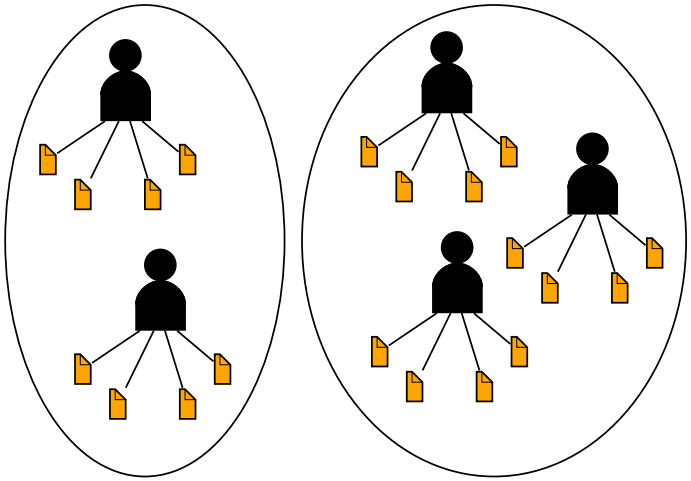
Shards

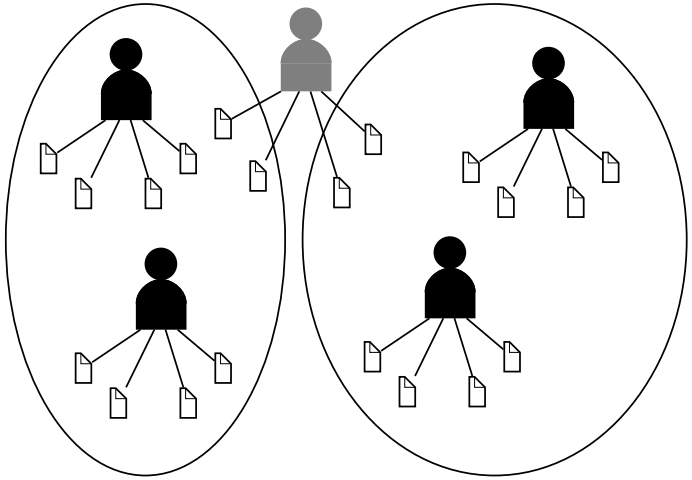




- Contributions – document icons
- Author – human icons
- Shard – ellipse
- Contributions of same author must be in the same shard!
- Defective hash function put them in different shards

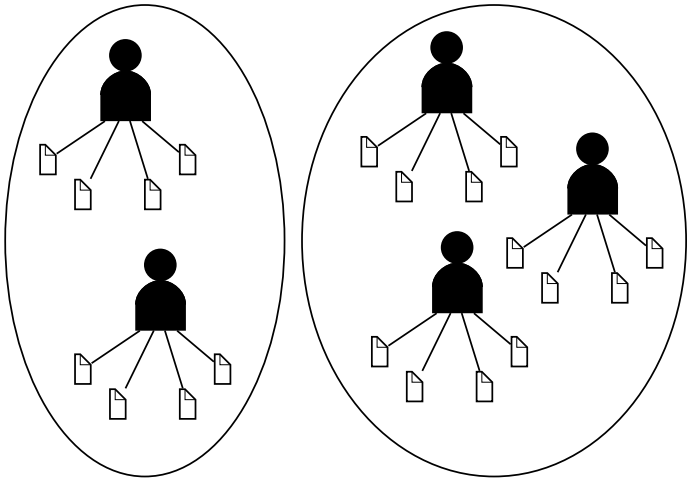
Shards

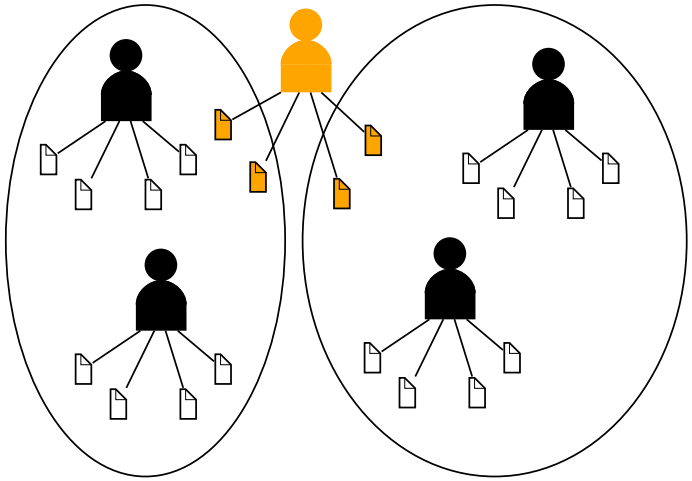




- Contributions – document icons
- Author – human icons
- Shard – ellipse
- Contributions of same author must be in the same shard!
- Defective hash function put them in different shards

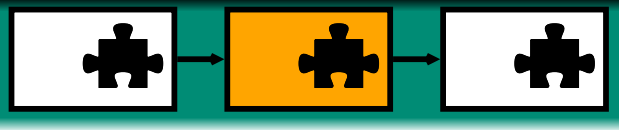
Shards





- Contributions – document icons
- Author – human icons
- Shard – ellipse
- Contributions of same author must be in the same shard!
- Defective hash function put them in different shards

Features



Time distance (continuous)

$$= \begin{cases} 0 & \text{year}(c_1) = \perp \\ & \vee\ \text{year}(c_2) = \perp \\ -1 & |\,\text{year}(c_1) - \text{year}(c_2)| > 70 \\ 1 - \left(\frac{\text{year}(c_1) - \text{year}(c_2)}{70}\right)^2 & \text{otherwise} \end{cases}$$

Time distance (discrete)

$$= \begin{cases} 0 & \text{year}(c_1) = \perp \vee \text{year}(c_2) = \perp \\ -1 & |\,\text{year}(c_1) - \text{year}(c_2)| > 70 \\ 1 & \text{otherwise} \end{cases}$$

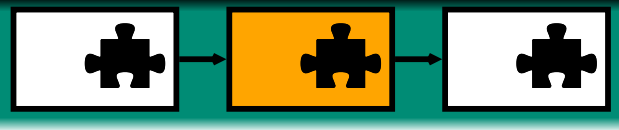
Journal

$$= \begin{cases} 0 & \text{journal}(c_1) = \perp \vee \text{journal}(c_2) = \perp \\ 1 & \text{journal}(c_1) = \text{journal}(c_2) \\ -0.1 & \text{otherwise} \end{cases}$$

Email

$$= \begin{cases} 0 & \text{email}(c_1) = \perp \vee \text{email}(c_2) = \perp \\ 1 & \text{email}(c_1) = \text{email}(c_2) \\ -0.1 & \text{otherwise} \end{cases}$$

Features



Language

$$= \begin{cases} 0 & \text{language}(c_1) = \perp \vee \text{language}(c_2) = \perp \\ 0.05 & \text{language}(c_1) = \textit{eng} \vee \text{language}(c_2) = \textit{eng} \\ 0.1 & \text{language}(c_1) = \text{language}(c_2) \\ -1 & \text{otherwise} \end{cases}$$

Keywords (discrete)

$$= \begin{cases} 0 & \text{keyword}(c_1) = \emptyset \vee \text{keyword}(c_2) = \emptyset \\ -1 & \frac{|\text{keyword}(c_1) \cap \text{keyword}(c_2)|}{|\text{keyword}(c_1) \cup \text{keyword}(c_2)|} < 0.25 \\ 1 & \text{otherwise} \end{cases}$$

Keywords (continuous)

$$= \begin{cases} 0 & \text{keyword}(c_1) = \emptyset \\ & \vee \text{keyword}(c_2) = \emptyset \\ \frac{|\text{keyword}(c_1) \cap \text{keyword}(c_2)|}{|\text{keyword}(c_1) \cup \text{keyword}(c_2)|} * 2 - 1 & \text{otherwise} \end{cases}$$

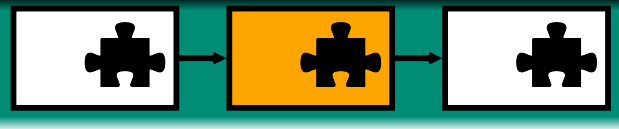
Self-citation

$$= \begin{cases} 1 & \textit{name}(c_1) = \textit{name}(\textit{reference}(c_1)) \\ 0 & \text{otherwise} \end{cases}$$

Co-authorship

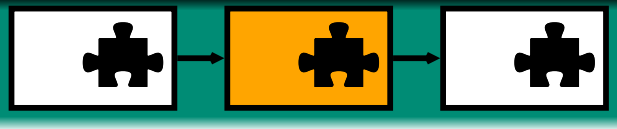
$$= \begin{cases} 0.7 & |\text{coauthors}(c_1) \cap \text{coauthors}(c_2)| = 1 \\ 1 & |\text{coauthors}(c_1) \cap \text{coauthors}(c_2)| > 1 \\ 0 & \text{otherwise} \end{cases}$$

Features' aspects



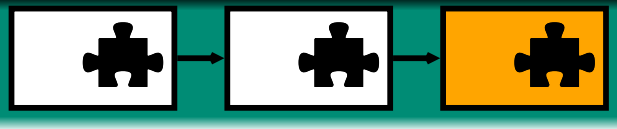
- **Discretization level**
  1. Discrete.
  2. Continuous.
- **Polarisation level**
  1. Polarised. Highly positive (negative) indicator and simultaneously weak negative (positive) indicator, e.g. e-mail or journal feature.
  2. Fair. A feature can be equally important as positive and negative indicator, e.g. discrete time distance features.
- **Structure**
  1. Flat structure.
  2. Graph structure.

Weights



- Some features can single-handedly prove that two contributions belong to the same author.
- Other features are only weak indicators (e.g. contributing to the same journal)
- The weight of a feature reflects the feature's impact on the name disambiguation process.

The clusterization function



- Goes according to an exchangeable clustering function
- Threshold
- Modification of single-linkage clustering

Work completed

- author name disambiguation framework
- basic feature set
- sketch of weights
- clusterization function

Future plans

- Feature generator (by a genetic programming or the Apriori algorithm)
- Weight estimator (the AdaBoost algorithm)
- Incremental version of the framework

Acknowledgements

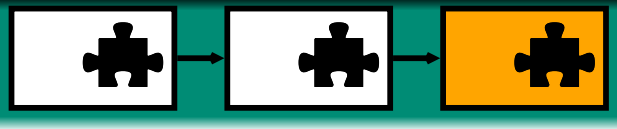
- EuDML project is partly financed by the European Union through its Competitiveness and Innovation Programme (Information and Communications Technologies Policy Support Programme, "Open access to scientific information", Grant Agreement no. 250,503).
- The authors would like to thank Zentralblatt MATH for providing their authority file for the purpose training and evaluation of our name disambiguation module

Thank you

Thank you!
Questions?

Piotr Jan Dendek
p.dendek@icm.edu.pl

The clusterization function - steps



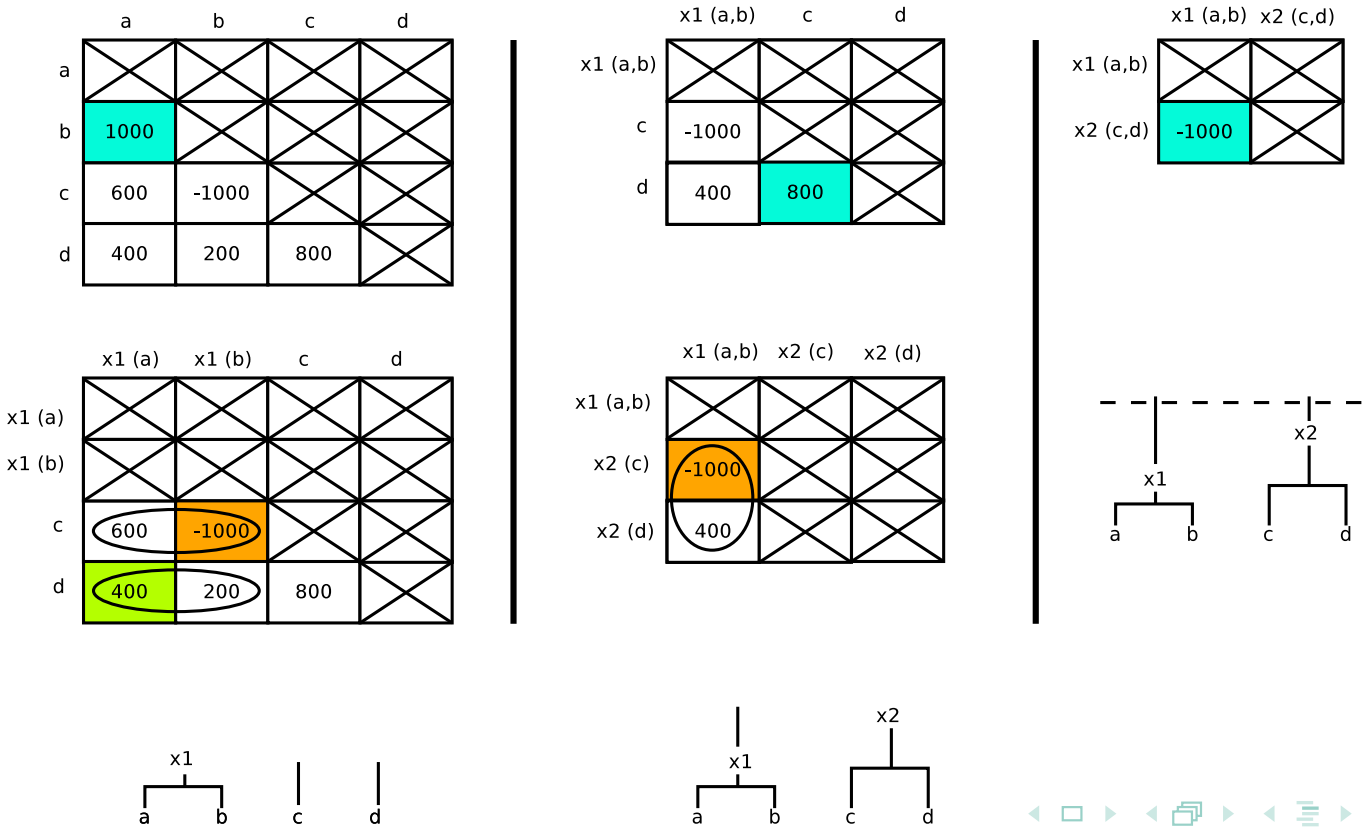
1. Take two “active” contribution clusters with top level score
2. If this score is below a given threshold – procedure ends.
3. Deactivate one of clusters.
4. Merge chosen clusters into one and recalculate new cluster affinities
5. Repeat step 1.

The clusterization function – equation

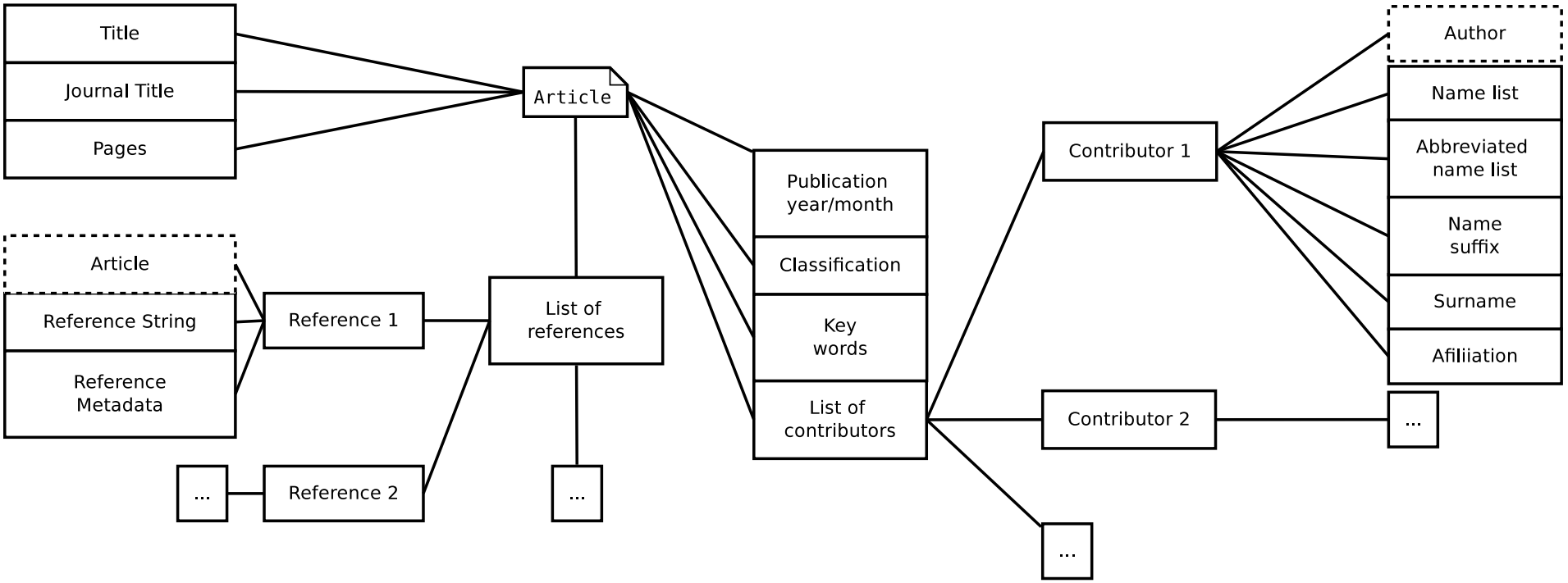
- Recalculation equation:

$$\forall_{1<i<N}\forall_{i\neq a}\forall_{i\neq b}\sigma(c_a, c_i) = \sigma(c_b, c_i) = \begin{cases} -\infty & \sigma(c_a, c_i) < T \\ & \vee\ \sigma(c_b, c_i) < T \\ \sigma(c_a, c_i) & \sigma(c_a, c_i) > \sigma(c_b, c_i) \\ \sigma(c_b, c_i) & \sigma(c_a, c_i) \leqslant \sigma(c_b, c_i) \end{cases}$$

- Example



Document metadata as a graph



Relations to be retrieved

- Information to be retrieved: