TOOLS AND TECHNIQUES

"FEEDING RECOMMENDER SYSTEMS WITH LINKED OPEN DATA"

Tommaso Di Noia

t.dinoia@poliba.it

Politecnico di Bari
ITALY

Recommender Systems

A definition

Recommender Systems (RSs) are *software tools* and *techniques providing suggestions for items to be of use to a user*.

[F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. **Recommender Systems Handbook**. Springer, 2011.]

Input Data:

A set of users $U=\{u_1, \ldots, u_M\}$

A set of items $X=\{x_1, \ldots, x_N\}$

The rating matrix $R=[r_{u,i}]$

Problem Definition:

Given user u and target item i

Predict the rating $r_{u,i}$


The rating matrix

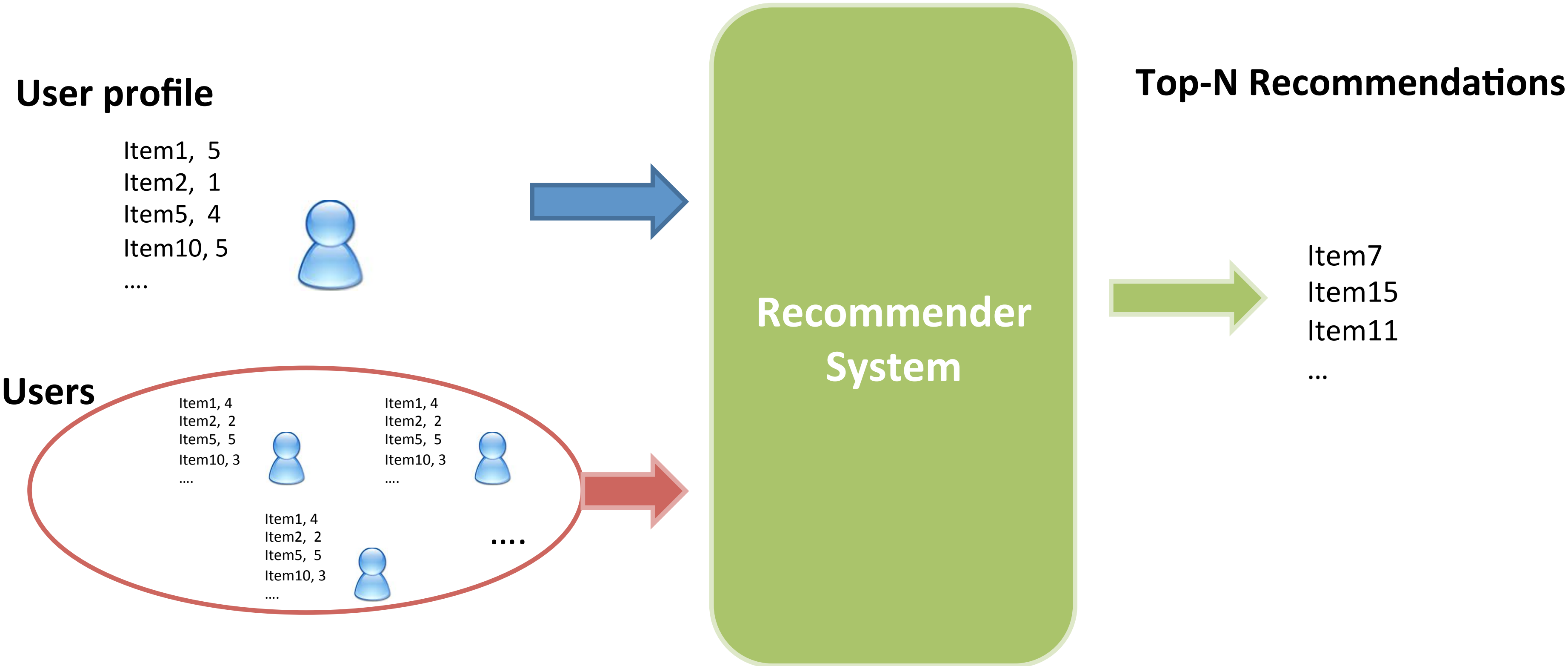
	The Matrix	Titanic	I love shopping	Argo	Love Actually	The hangover
Tommaso	5	1	2	4	3	**??**
Enrico	2	4	5	3	5	2
Sean	4	3	2	4	1	3
Natasha	3	5	1	5	2	4
Valentina	4	4	5	3	5	2

The rating matrix (in the real world)

	The Matrix	Titanic	I love shopping	Argo	Love Actually	The hangover
Tommaso	5			4	3	**??**
Enrico	2	4	5		5	
Sean		3		4		3
Natasha	3	5		5	2	
Valentina	4	4	5		5	2

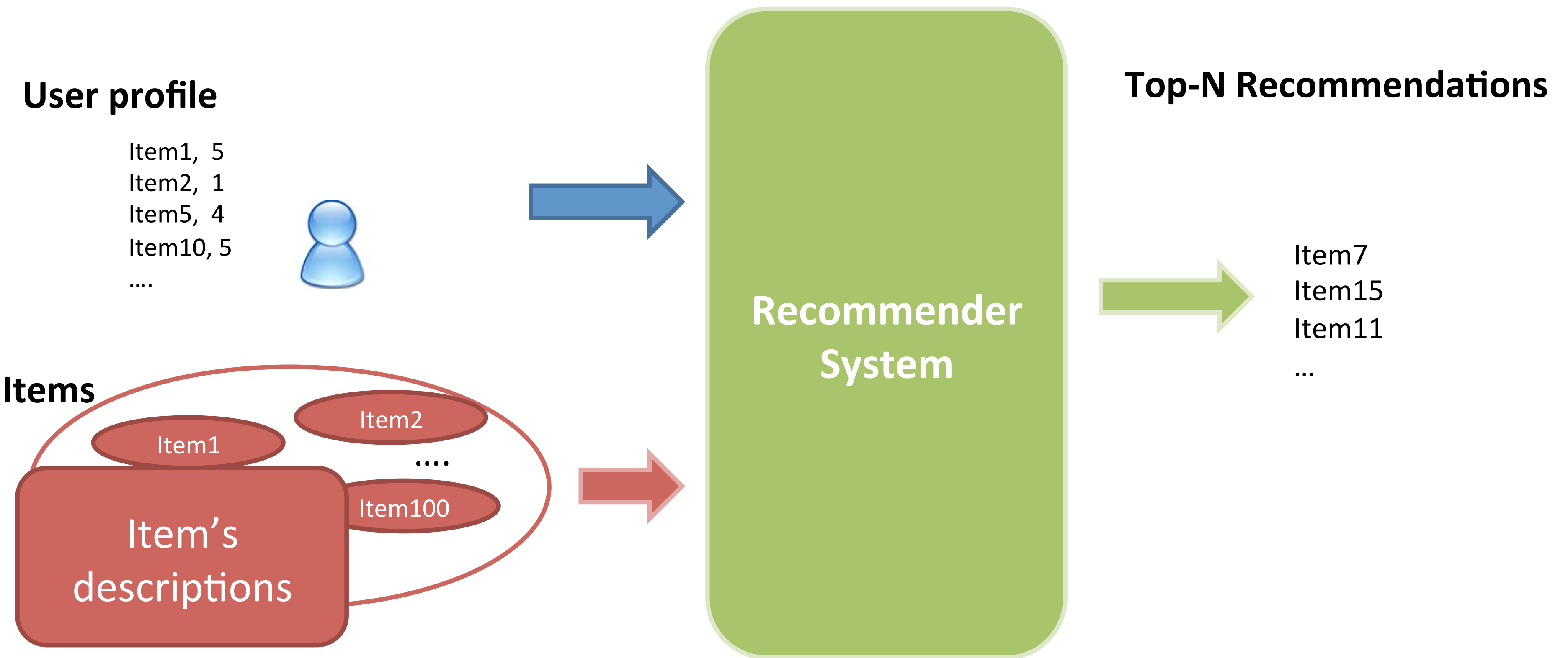
Collaborative Recommender Systems

Collaborative RSs recommend items to a user by identifying other users with a similar profile



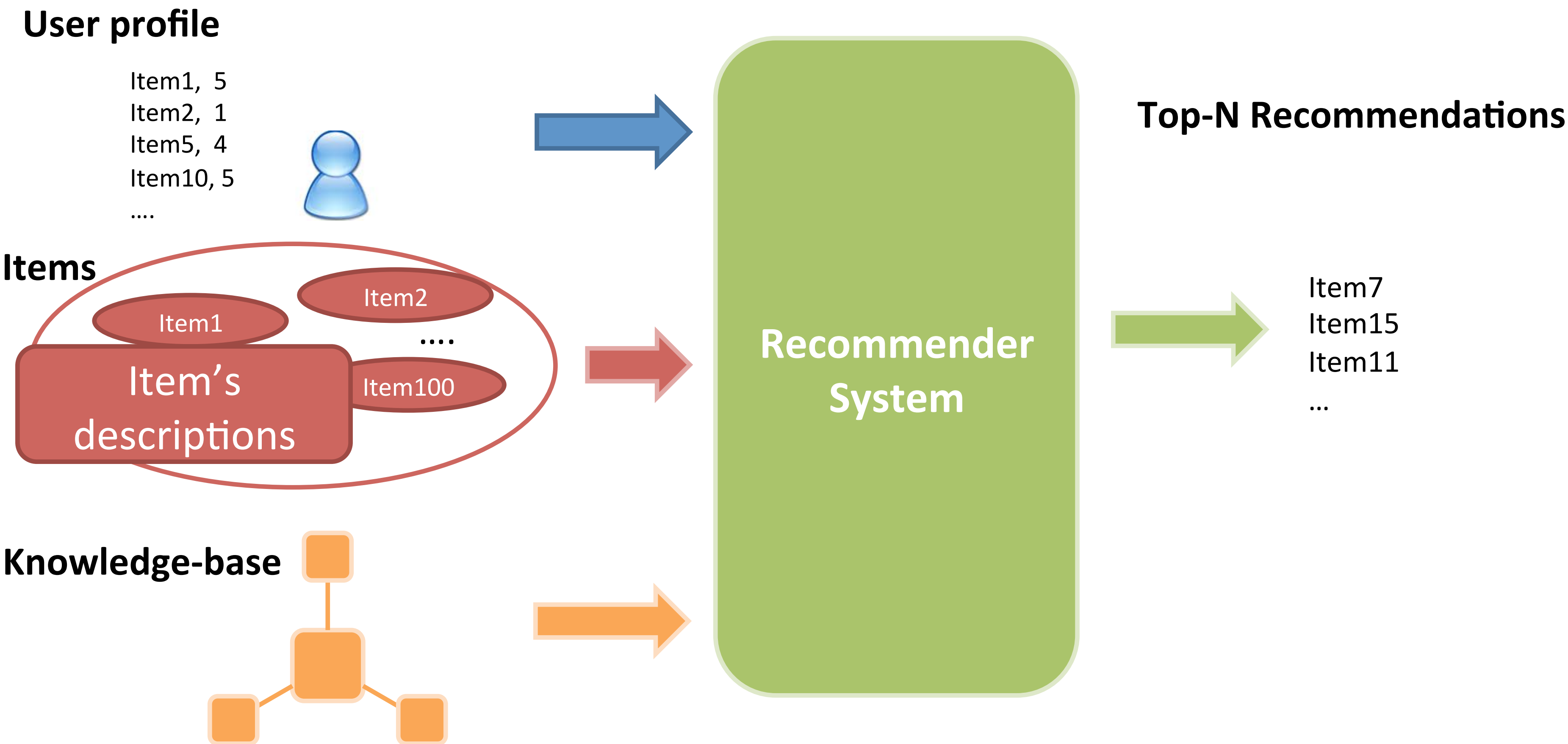
Content-based Recommender Systems

CB-RSs recommend items to a user based on their description and on the profile of the user's interests

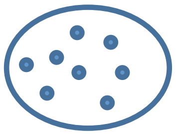


Knowledge-based Recommender Systems

KB-RSs recommend items to a user based on their description and domain knowledge encoded in a knowledge base



User-based Collaborative Recommendation

 $= X$

Pearson's correlation coefficient

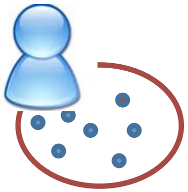
$$sim(u_i, u_j) = \frac{\sum_{x \in X} (r_{u_i,x} - \bar{r}_{u_i}) * (r_{u_j,x} - \bar{r}_{u_j})}{\sqrt{\sum_{x \in X} (r_{u_i,x} - \bar{r}_{u_i})^2} * \sqrt{\sum_{x \in X} (r_{u_j,x} - \bar{r}_{u_j})^2}}$$

	The Matrix	Titanic	I love shopping	Argo	Love Actually	The hangover
Tommaso	5	1	2	4	3	**??**
Enrico	2	4	5	3	5	2
Sean	4	3	2	4	1	3
Natasha	3	5	1	5	2	4
Valentina	4	4	5	3	5	2

Rate prediction

$$r(u_i, x') = \bar{r}_{u_i} + \sum_{u_j} sim(u_i, u_j) * (r_{u_i}$$

Item-based Collaborative Recommendation

 $= X_{u_i}$

Cosine Similarity

$$sim(x_i, x_j) = \frac{x_i \cdot x_j}{|x_i| * |x_j|} = \frac{\sum_{u} r_{u,x_i} * r_{u,x_j}}{\sqrt{\sum_{u} r_{u,x_i}^2} * \sqrt{\sum_{u} r_{u,x}^2}}$$

Adjusted Cosine Similarity

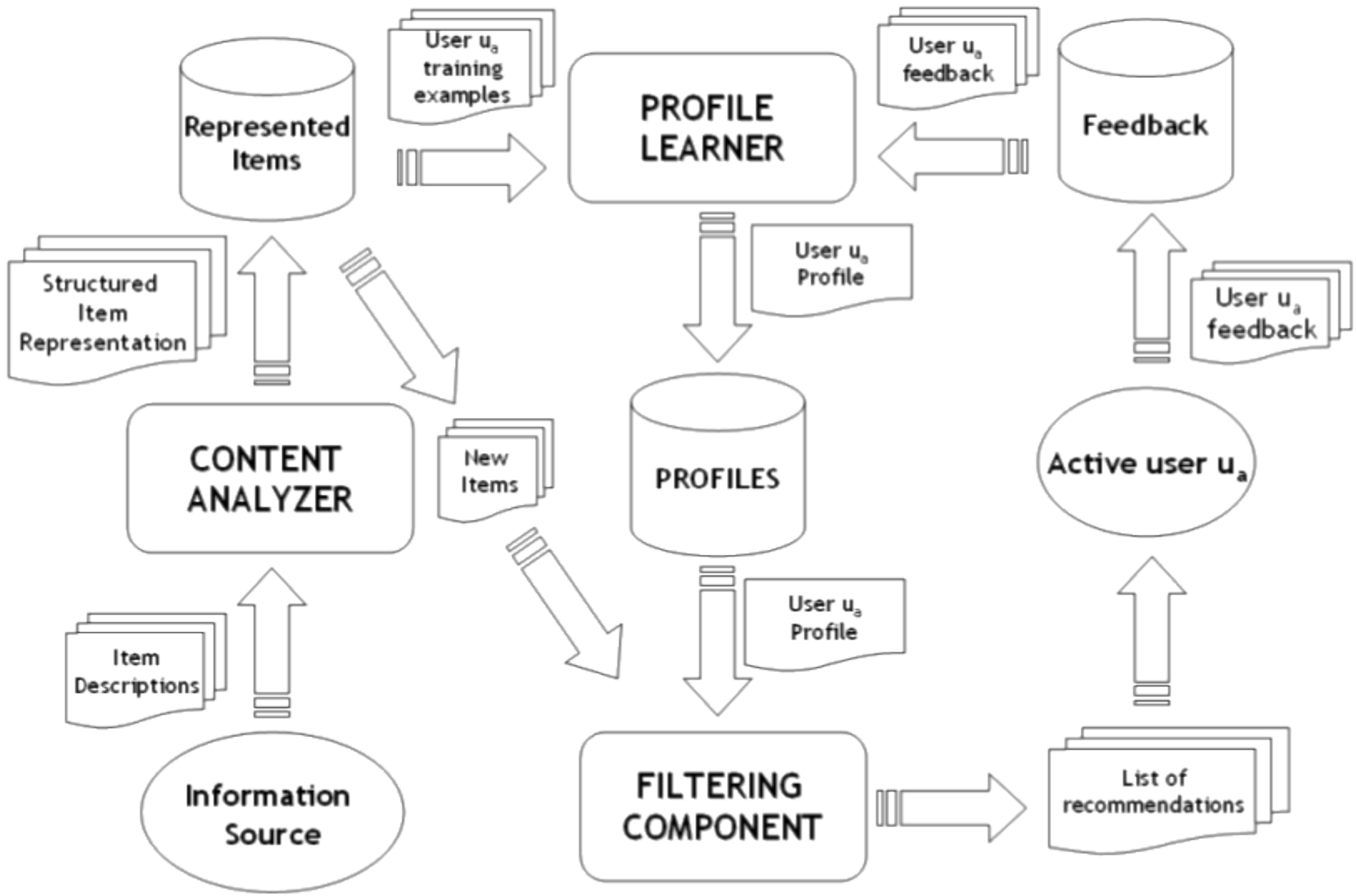
$$sim(x_i, x_j) = \frac{\sum_{u} (r_{u,x_i} - r_u) * (r_{u,x_j} - r_u)}{\sqrt{\sum_{u} (r_{u,x_i} - r_u)^2} * \sqrt{\sum_{u} (r_{u,x_j} - r_u)^2}}$$

	The Matrix	Titanic	I love shopping	Argo	Love Actually	The hangover
Tommaso	5	1	2	4	3	**??**
Enrico	2	4	5	3	5	2
Sean	4	3	2	4	1	3
Natasha	3	5	1	5	2	4
Valentina	4	4	5	3	5	2

Rate prediction

$$r(u_i, x') = \sum_{x \in X_{u_i}} sim(x, x'$$

Content-Based Recommender Systems



Content-Based Recommender Systems

- Items are described in terms of attributes/ features
- A finite set of values is associated to each feature
- Items representation is a (Boolean) vector

Content-Based Recommender Systems

- Compute similarity between items

Jaccard similarity

$$sim(x_i, x_j) = |x_i \cap x_j| / |x_i \cup x_j|$$

Content-Based Recommender Systems

- Compute similarity between items

Cosine similarity

$$sim(x_i, x_j) = x_i \cdot x_j / |x_i| * |x_j|$$

Content-Based Recommender Systems

- Compute similarity between items

Cosine similarity and TF-IDF

$$TF(v,x) = \#\ v \text{ appears in the description of } x$$

$$IDF(v) = \log |X| / \#\ v \text{ appears in } X$$

$$x = (TF(v_1, x) * IDF(v_1), \ldots, TF(v_n, x) * IDF(v_n))$$

$$sim(x_i, x_j) = x_i \cdot x_j / |x_i| * |x_j|$$

Content-Based Recommender Systems

Rate prediction

$$r(u_i, x') = \sum_{x \in X_{u_i}} sim(x, x') * r_{x,u_i} \Big/ \sum_{x \in X_{u_i}} sim(x, x'$$

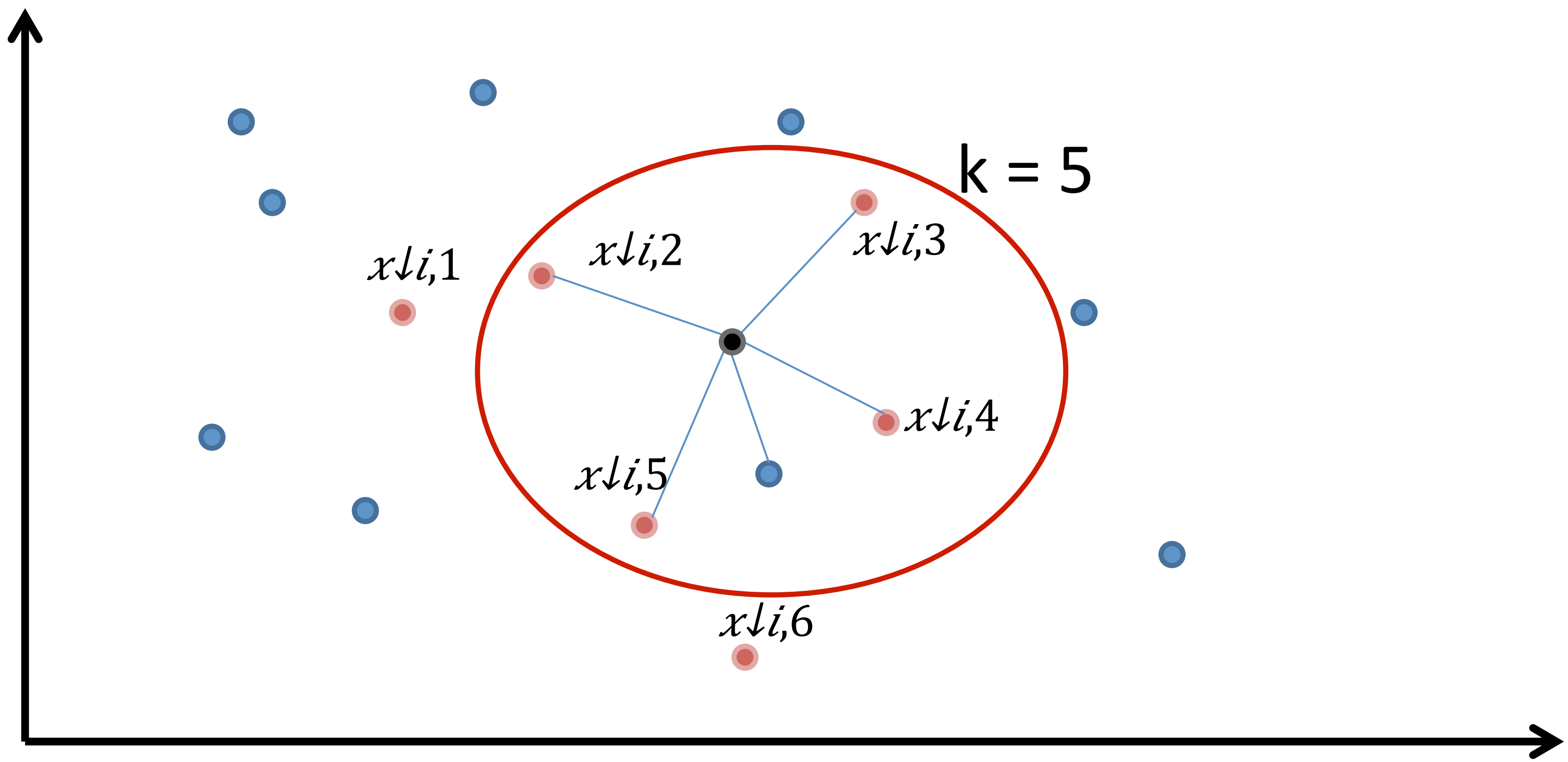
$$=$$

Content-Based Recommender Systems

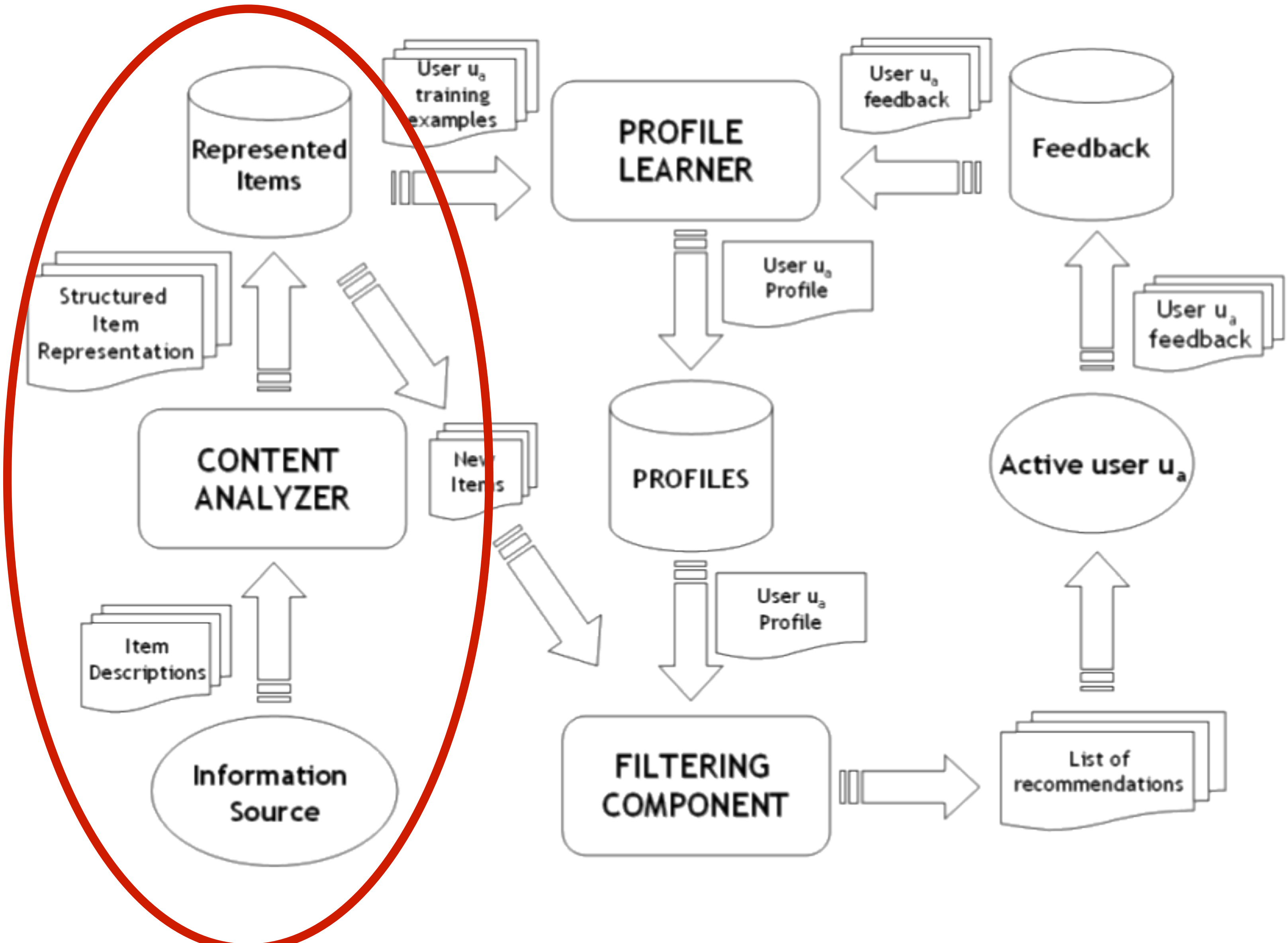
- Nearest neighbors
  - Given a set of items representing the user profile, select their most similar items which are not in the user profile
  - Predict the rate only for the N nearest neighbors

Content-Based Recommender Systems

- Nearest neighbors with kNN



Content-Based Recommender Systems



Main CB RSs Drawback: Limited Content Analysis

No suggestion is available if the analyzed content does not contain enough information to discriminate items the user might like from items the user might not like.*

The quality of CB recommendations are correlated with the quality of the features that are explicitly associated with the items.

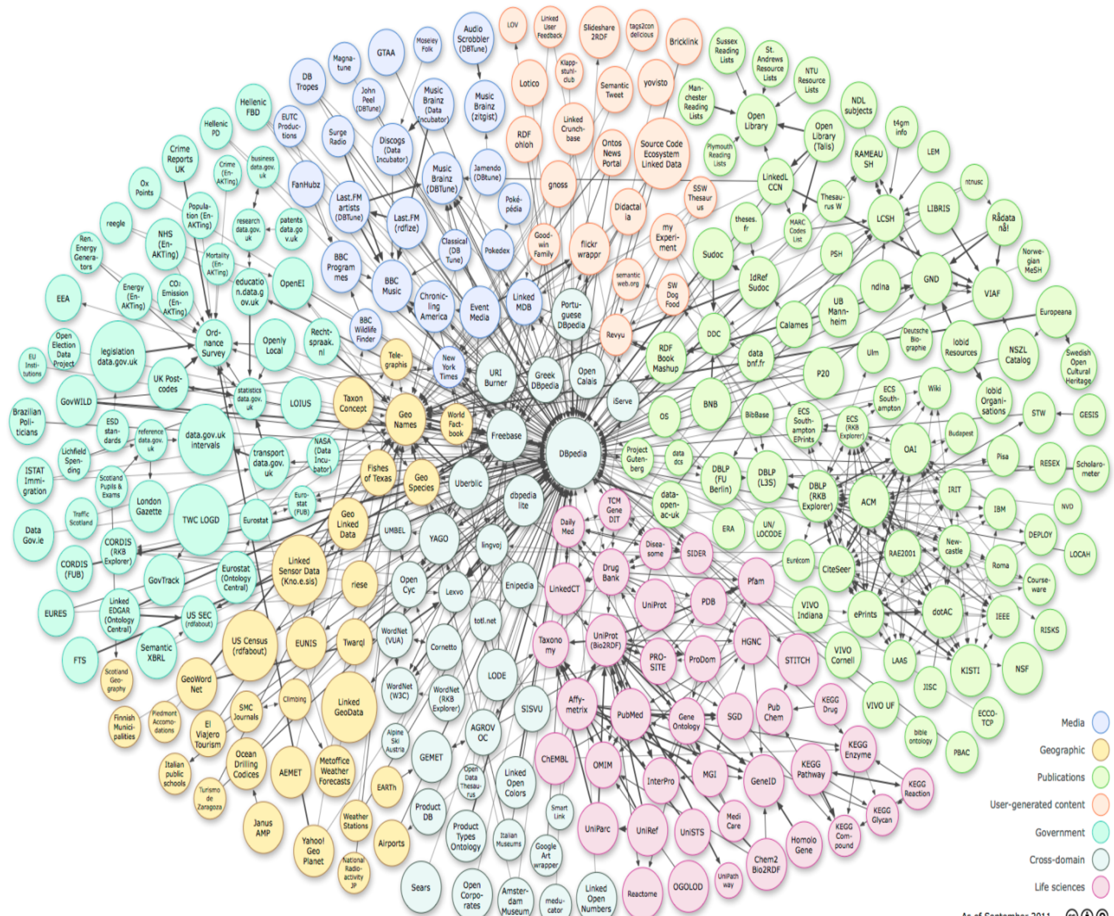
Need of domain knowledge!
We need rich descriptions of the items!

(*) P. Lops, M. de Gemmis, G. Semeraro. Content-based Recommender Systems: State of the Art and Trends. In: P. Kantor, F. Ricci, L. Rokach and B. Shapira, editors, Recommender Systems Handbook: A Complete Guide for Research Scientists & Practitioners

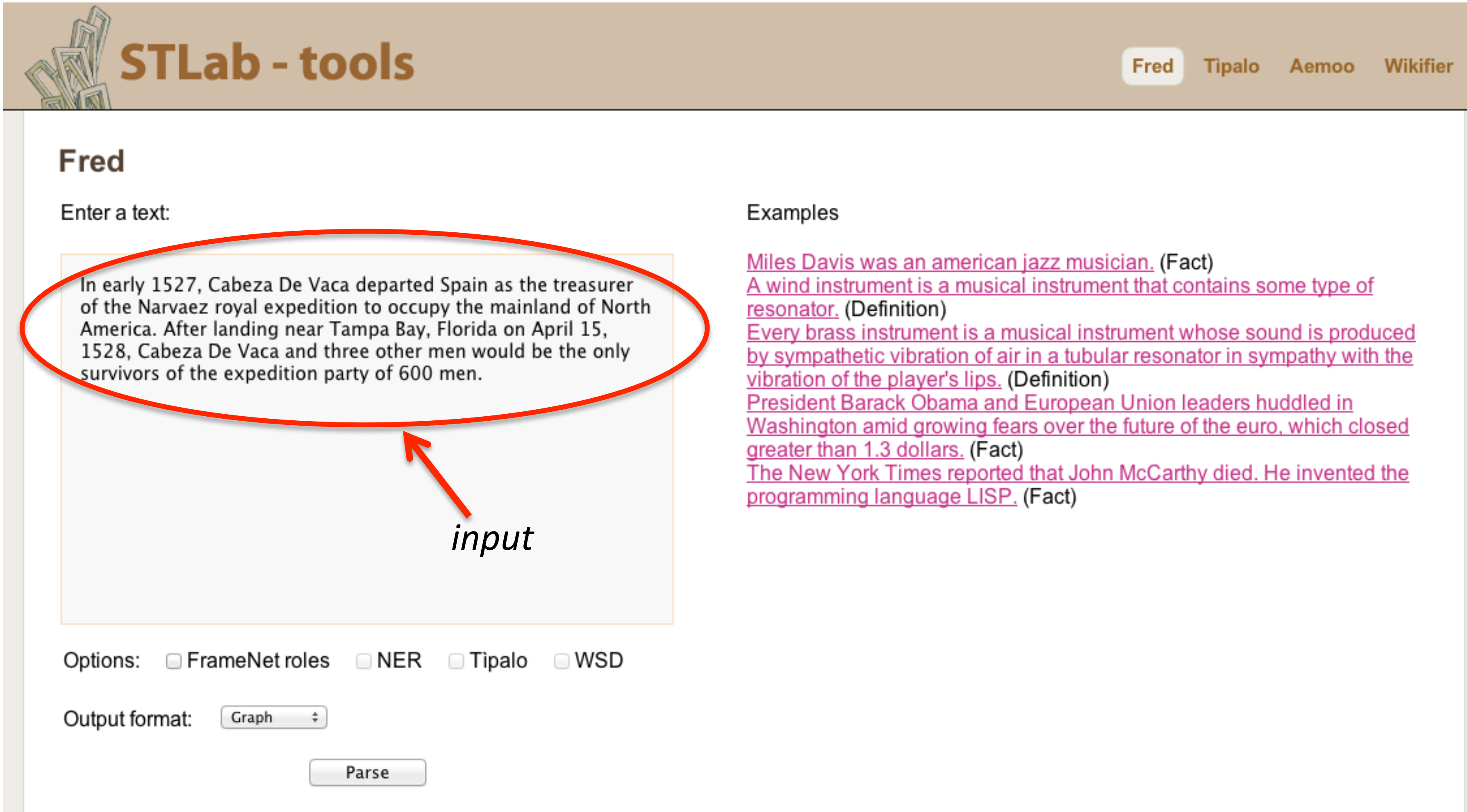
A solution based on Linked Data

Use Linked Data to mitigate the limited content analysis issue

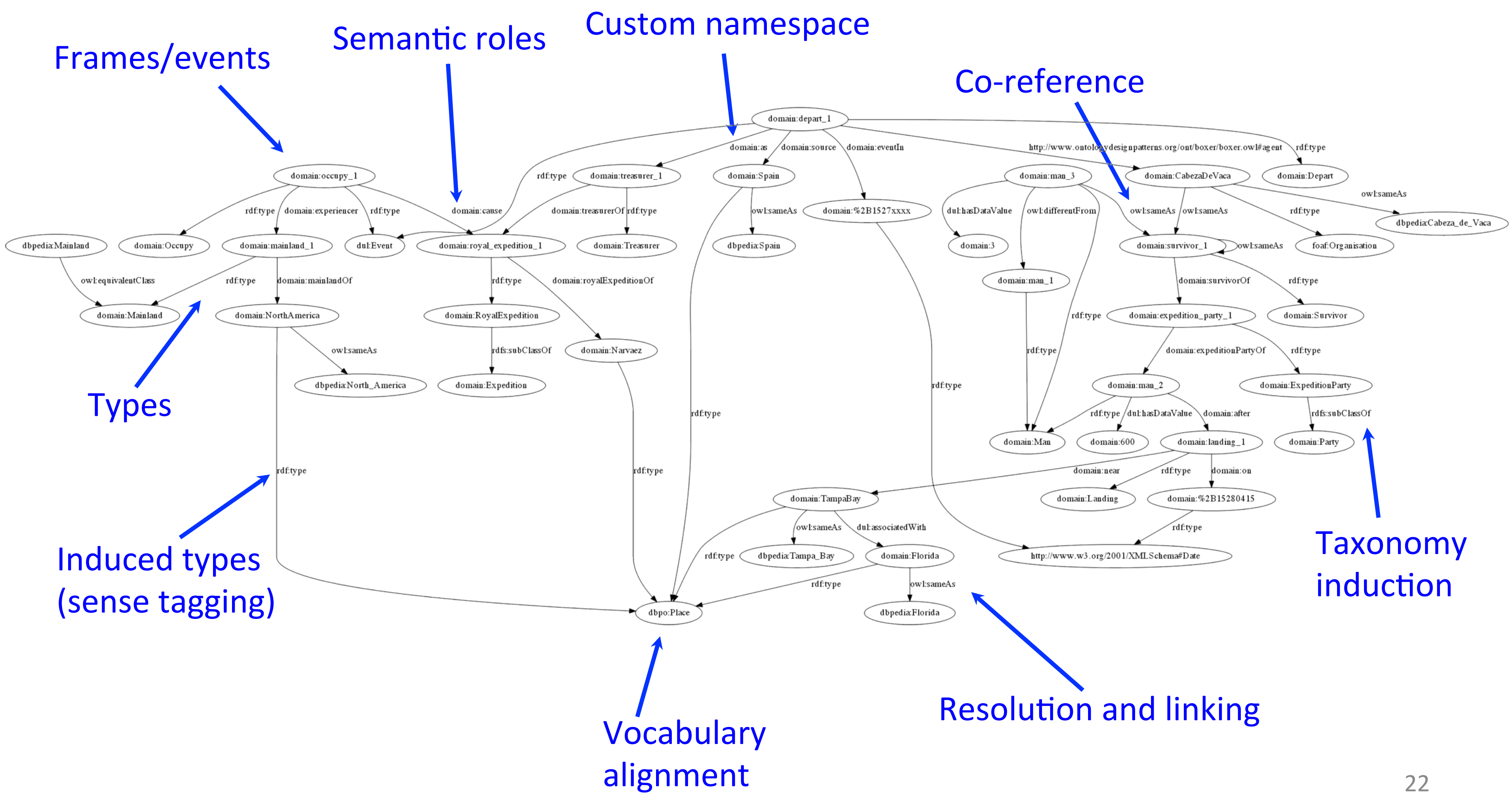
- **Plenty of structured data available**
- **No Content Analyzer required**



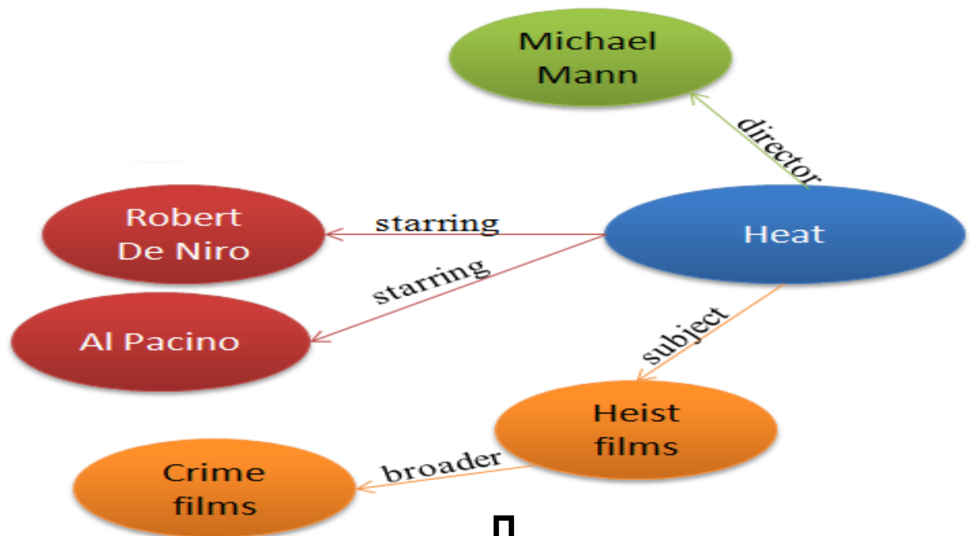
FRED: Transforming Natural Language text to RDF/OWL graphs



FRED: RDF/OWL graph output



Linked Data as structured information source for items descriptions



Rich items descriptions

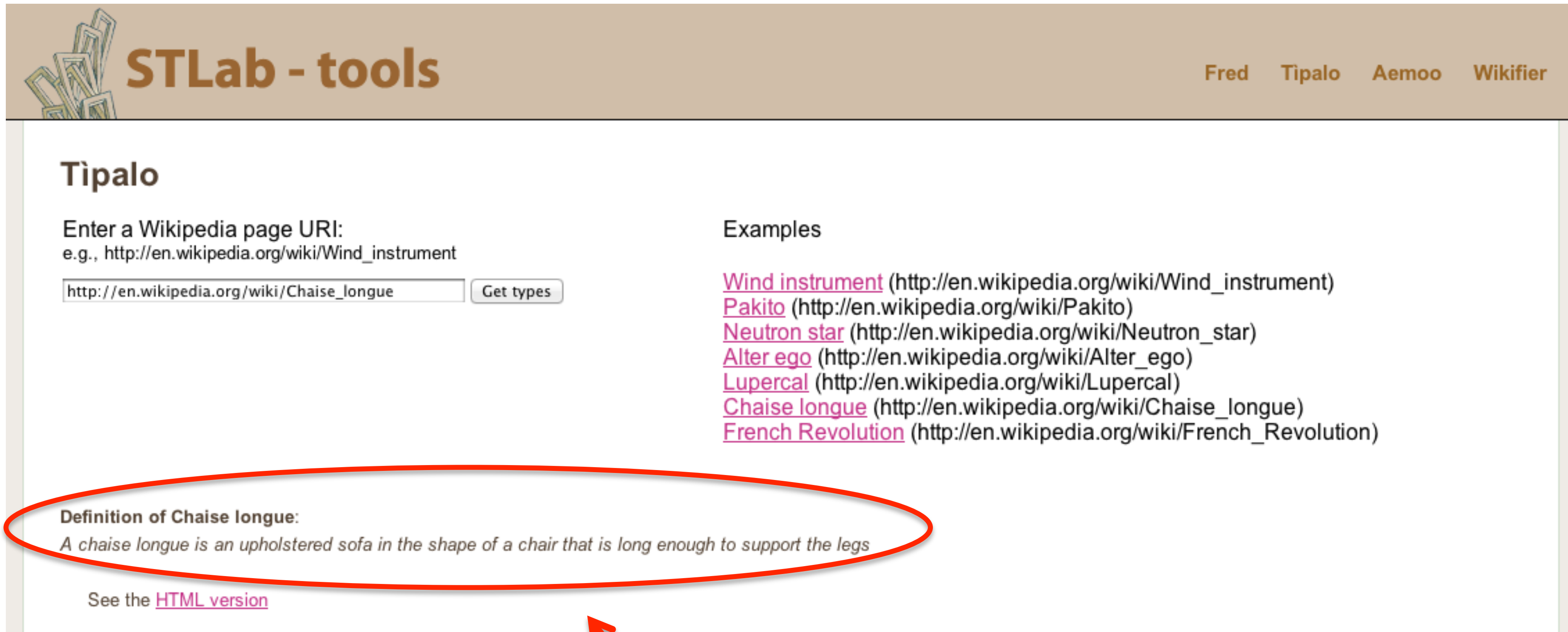
	STARRING		DIRECTOR	SUBJECT+BROADER	
Heat	Robert DeNiro	Al Pacino	Michael Mann	Heist films	Crime films

Select the domain(s) of your RS

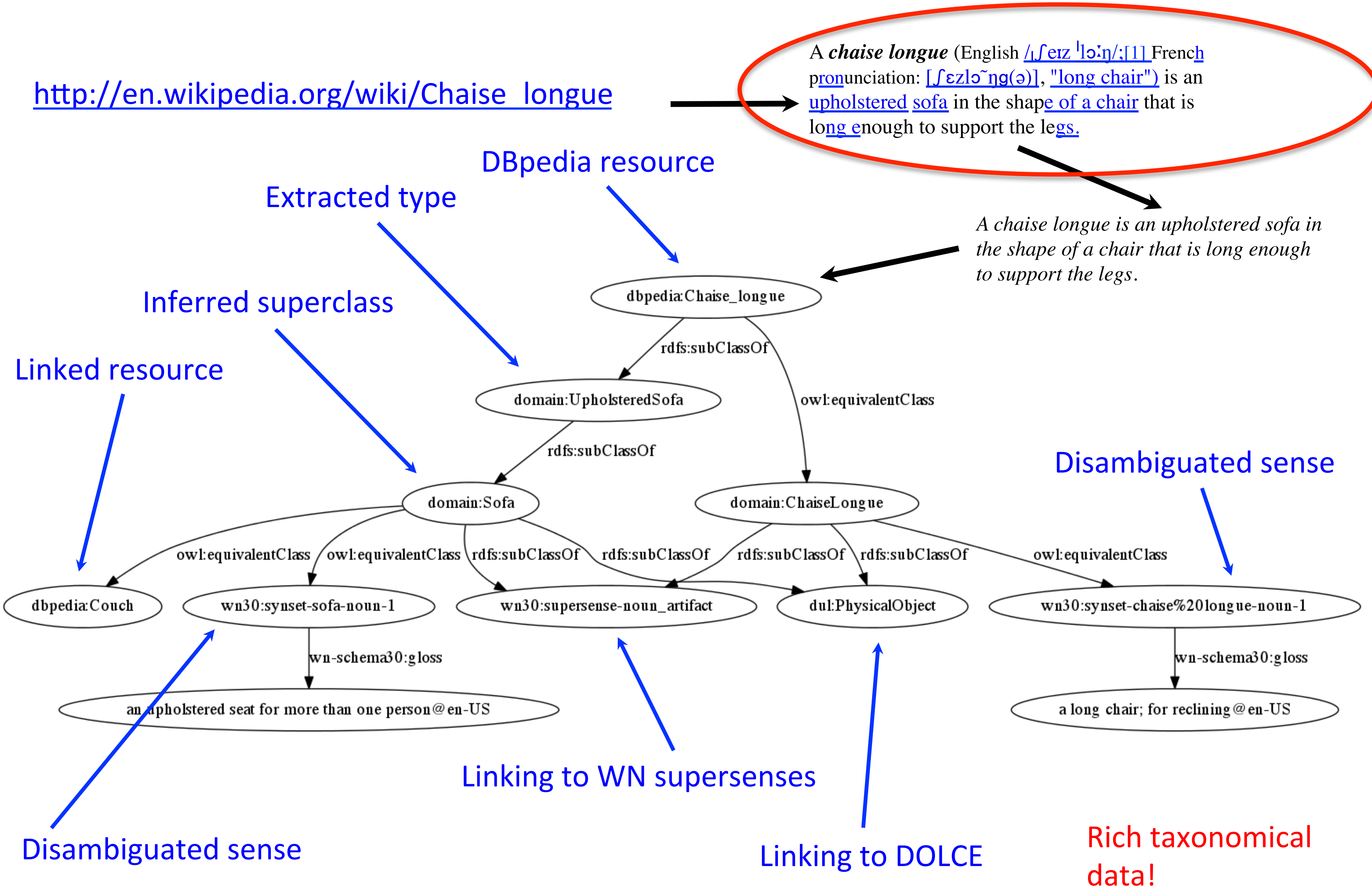
```
SELECT count(?i) AS ?num ?c
WHERE {
   ?i a ?c .
   FILTER(regex(?c, "^http://dbpedia.org/ontology")) .
}
ORDER BY DESC(?num)
```

num	c
956476	http://dbpedia.org/ontology/Agent
763644	http://dbpedia.org/ontology/Person
572728	http://dbpedia.org/ontology/Place
387166	http://dbpedia.org/ontology/PopulatedPlace
348520	http://dbpedia.org/ontology/Settlement
333270	http://dbpedia.org/ontology/Work
277476	http://dbpedia.org/ontology/OrganisationMember
277476	http://dbpedia.org/ontology/SportsTeamMember

Tìpalo: automatic typing of DBpedia entities



Input: natural language definition of an entity



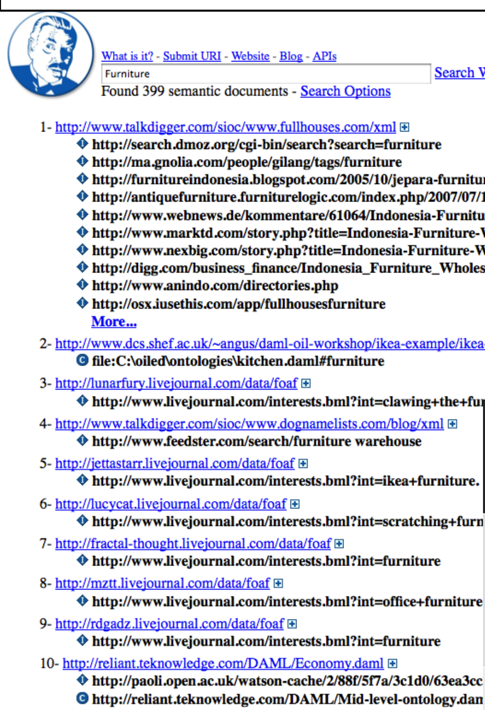
http://en.wikipedia.org/wiki/Chaise_longue
A *chaise longue* (English /ˌʃeɪz ˈlɔːŋ/;[1] French pronunciation: [ʃɛzlɔ˜ŋg(ə)], "long chair") is an upholstered sofa in the shape of a chair that is long enough to support the legs.
A chaise longue is an upholstered sofa in the shape of a chair that is long enough to support the legs.
DBpedia resource
Extracted type
Inferred superclass
Linked resource
Disambiguated sense
dbpedia:Chaise_longue
rdfs:subClassOf
owl:equivalentClass
domain:UpholsteredSofa
rdfs:subClassOf
domain:Sofa
domain:ChaiseLongue
owl:equivalentClass
owl:equivalentClass
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
owl:equivalentClass
dbpedia:Couch
wn30:synset-sofa-noun-1
wn30:supersense-noun_artifact
dul:PhysicalObject
wn30:synset-chaise%20longue-noun-1
wn-schema30:gloss
wn-schema30:gloss
an upholstered seat for more than one person@en-US
a long chair; for reclining@en-US
Linking to WN supersenses
Disambiguated sense
Linking to DOLCE
Rich taxonomical data!

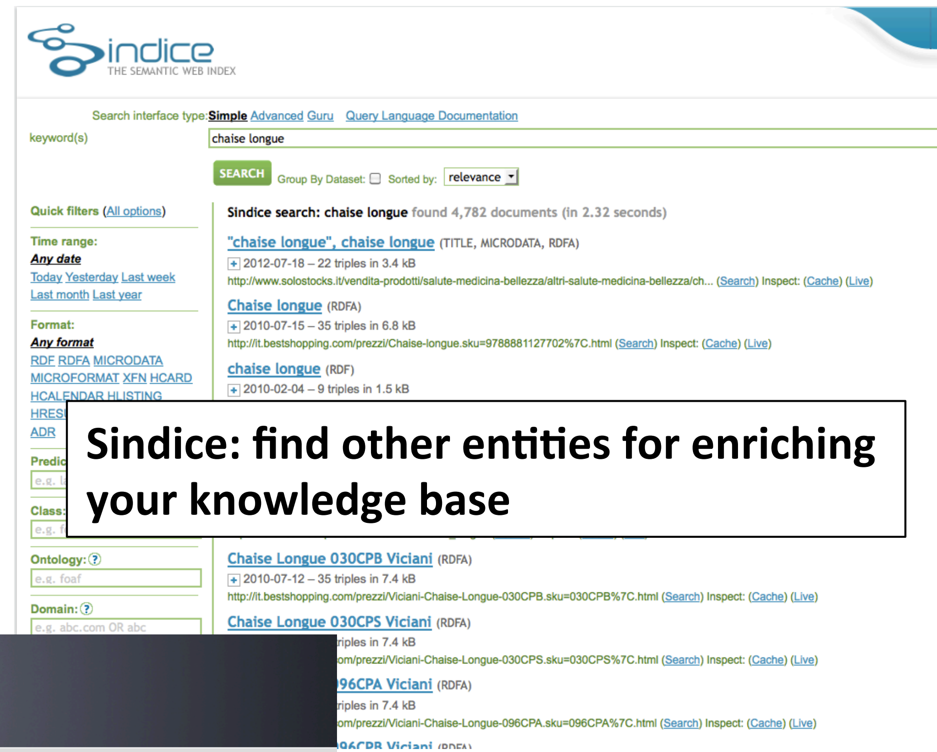
Select the sub-graph you are interested in

```
select ?m ?p ?o1
where{
  ?m ?p ?o1 .
  ?m dcterms:subject ?o.
  dbpedia:The_Matrix dcterms:subject ?o.
  ?m a dbpedia-owl:Film.
  filter(regex(?p,"^http://dbpedia.org/ontology/")).
}
order by ?m
```

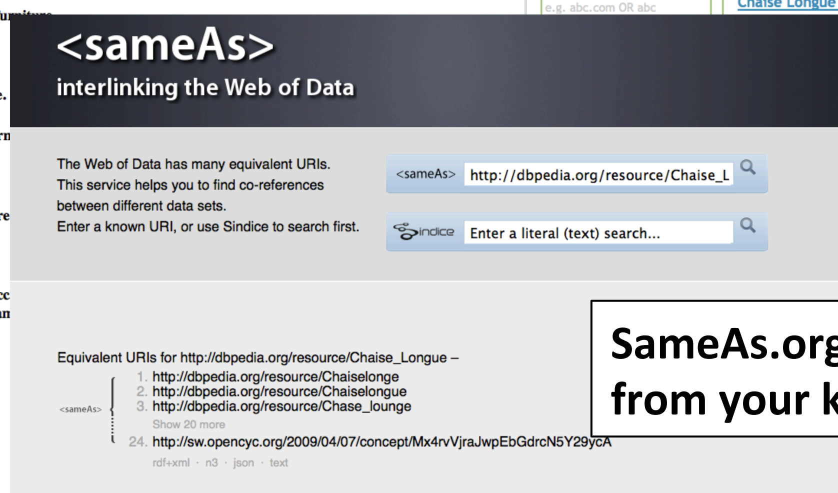
Enriching Annotations with Resources From the Web

Watson: find ontologies on the Semantic Web for enriching the representation of your data



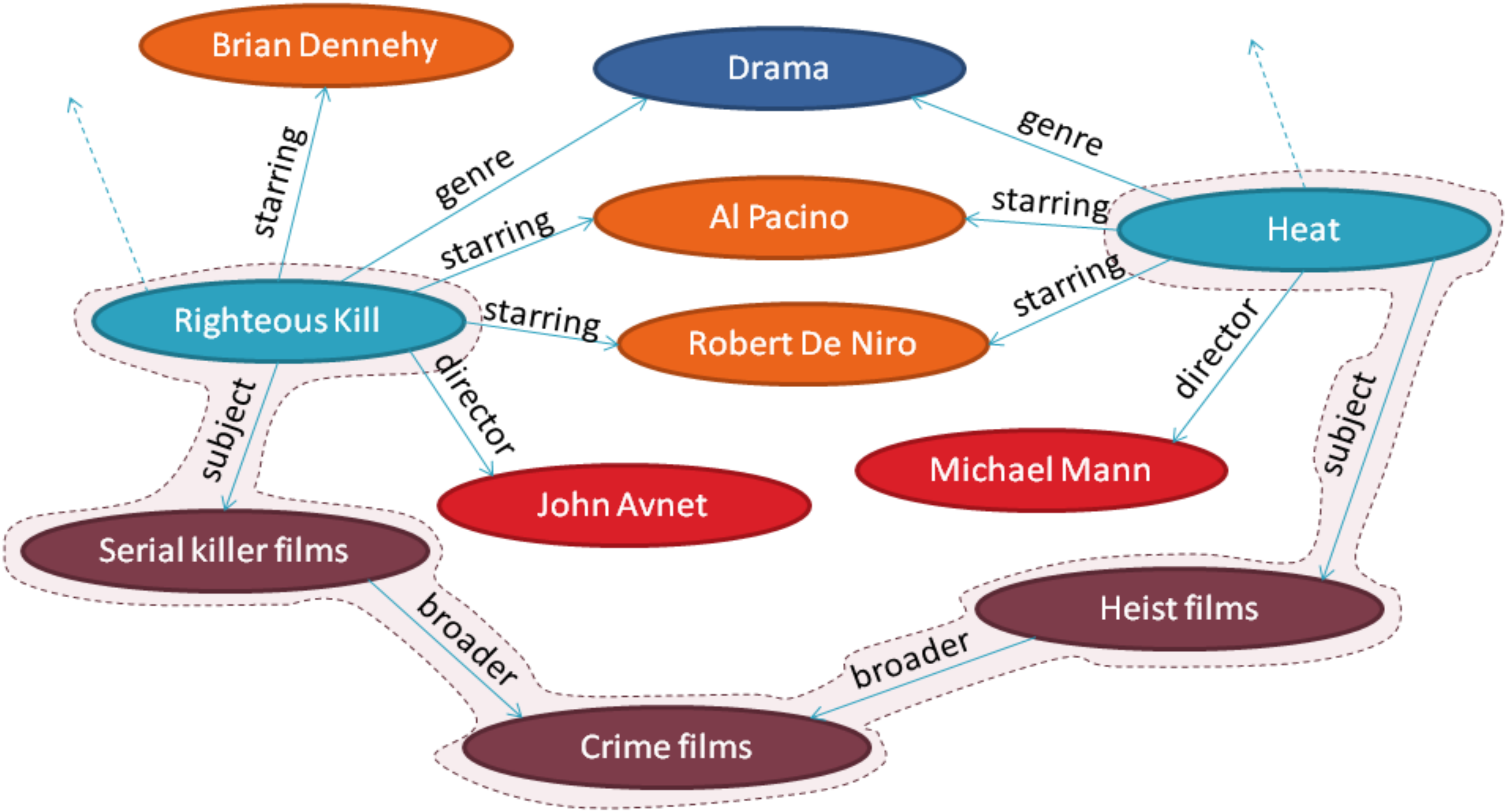


Sindice: find other entities for enriching your knowledge base

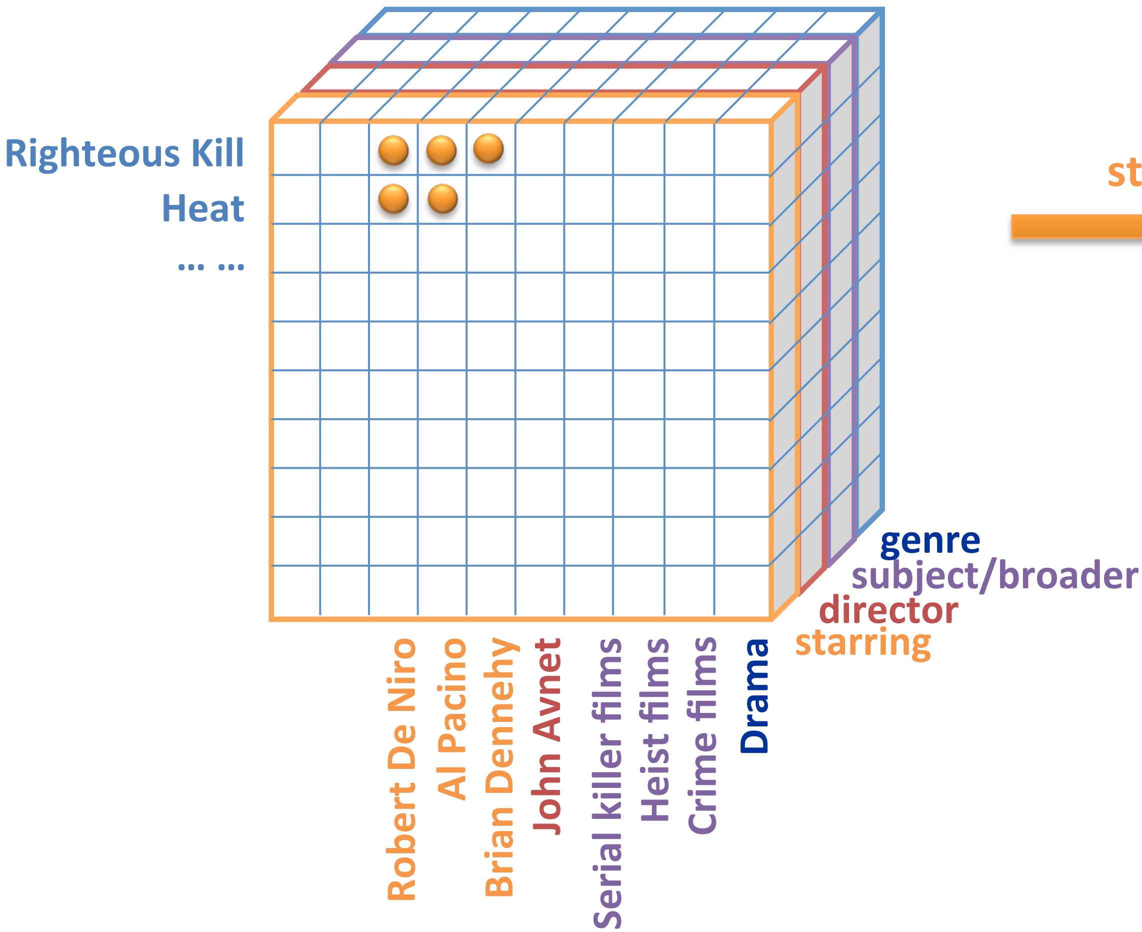


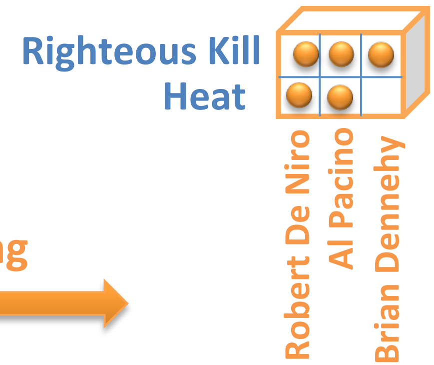
SameAs.org: find other entities to link to from your knowledge base

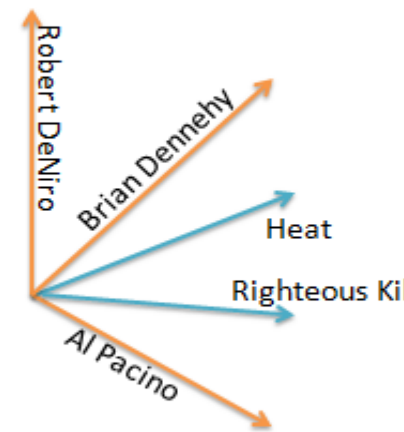
Computing similarity in LOD datasets



Vector Space Model for LOD







Vector Space Model for LOD

Righteous Kill

<u>STARRING</u>	Al Pacino (v1)	Robert De Niro (v2)	Brian Dennehy (v3)
Righteous Kill (m1)	X	X	X
Heat (m2)	X	X	

Heat

$$w_{AlPacino,Heat} = tf_{AlPacino,Heat} * idf_{AlPacino}$$

Righteous Kill (x1)	$w_{v1,x1}$	$w_{v2,x1}$	$w_{v3,x1}$
Heat (x2)	$w_{v1,x2}$	$w_{v2,x2}$	0

Vector Space Model for LOD

Righteous Kill

STARRING	Al Pacino (a1)	Robert De Niro (a2)	Brian Dennehy (a3)
Righteous Kill (x1)	O	O	O
Heat (x2)	O	O	

Heat

$tf \in \{0,1\}$

$$w_{AlPacino,Heat} = tf_{AlPacino,Heat} * idf_{AlPacino}$$

Righteous Kill (m1)	$w_{a1,m1}$	$w_{a2,m1}$	$w_{a3,m1}$
Heat (m2)	$w_{a1,m2}$	$w_{a2,m2}$	0

Vector Space Model for LOD

$sim_{starring}(x_i, x_j) = w_{v_1, x_i} * w_{v_1, x_j} + w_{v_2, x_i} * w_{v_2, x_j} + w$

$\alpha_{starring} * sim_{starring}(x_i, x_j)$ +

$\alpha_{director} * sim_{director}(x_i, x_j)$ +

$\alpha_{subject} * sim_{subject}(x_i, x_j)$ +

...

=

$sim_{starring}(x_i, x_j)$

LOD-based CB RS

Given a user profile defined as: $\boldsymbol{X_{u_i} = \{ < x_i, r_{x_i, u_i} > \}}$

We can predict the rating using a ***Nearest Neighbor Classifier*** wherein the similarity measure is a linear combination of ***local property similarities***

$$r(u_i, x') = \sum_{< x_i, r_{x_i, u_i} > \in X_{u_i}} \frac{\sum_p \alpha_p * sim_p(x_i, x')}{P} * r_{x_i, u_i} / |X_{u_i}|$$

$$=$$

Missing in this presentation (not a complete list)

- Learning $\alpha \downarrow p$
- Explanation
  - Quite straight with a content-based approach
- Hybrid approaches
  - Mix a content based approach with a collaborative one
  - Collaborative similarity as a further vector space (property) of the content based approach?
  - Exploit the graph-based nature of both the rating matrix and the RDF graph
- …

Thank you!

Many thanks to **Vito Claudio Ostuni** and **Roberto Mirizzi**