Combiner approche logique et numérique pour la réconciliation de données et l'alignement d'ontologies.

Marie-Christine Rousset

Rémi Tournaire (*) et Alexandre Termier

Jean-Marc Petit

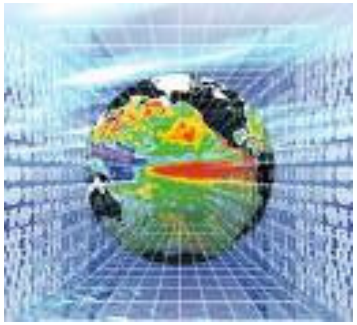
Fatiha Sais (*) et

Nathalie Pernelle

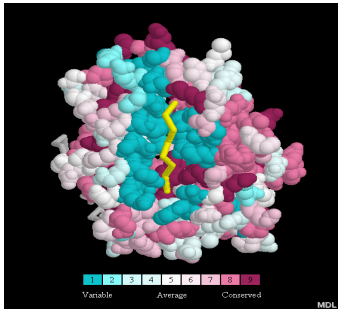
(*) auteur de certains transparents de cet exposé

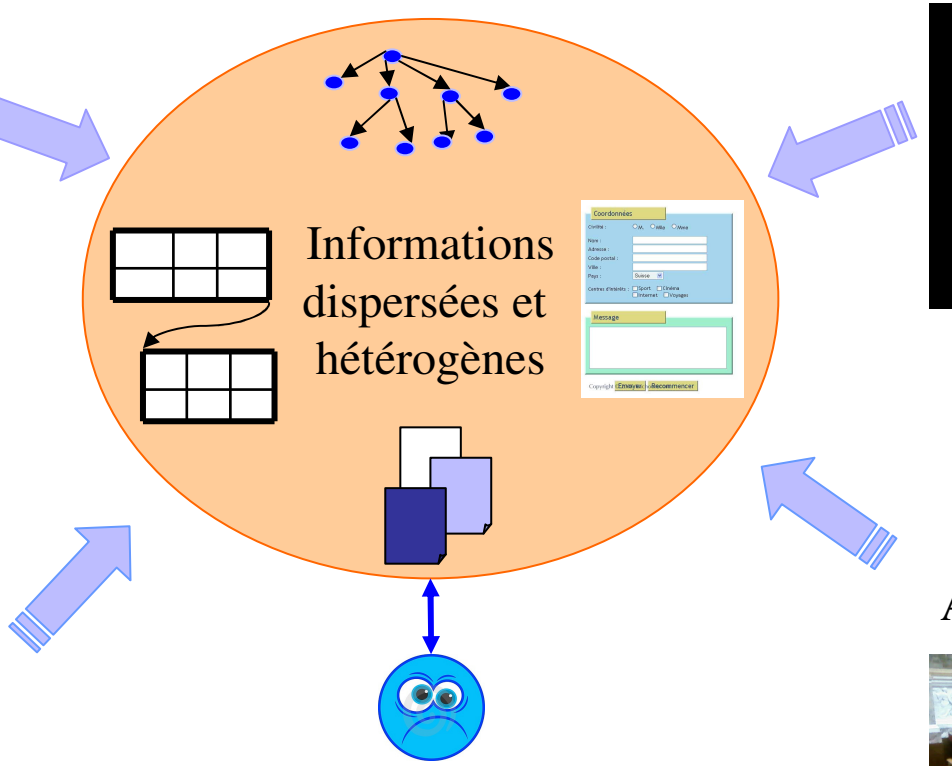
2 problèmes au cœur de l'intégration d'informations

Web



Sciences





Entreprises

Administrations

Alignement de schémas ou d'ontologies

\+ réconciliation de données

De nombreux travaux existants

- Beaucoup d'approches numériques, souvent empiriques:
  - calcul de similarités
    - entre tuples, entre attributs, entre relations, entre classes
- Peu d'approches logiques
  - S-Match
  - Découverte de mappings par composition
- Très peu d'approches combinant les deux

  $\Rightarrow$ Des résultats numériques déconnectés des contraintes logiques existantes sur les données à mettre en correspondance

  $\Rightarrow$ Difficulté d'interpréter ces résultats numériques en terme de correspondance logique

  équivalence, subsomption, ou recouvrement entre classes ?

Dénominateur commun de notre approche

- Exploiter la sémantique
  - en se fondant sur des formalismes mathématiques complémentaires
    - la logique, les systèmes d'équations ou les probabilités
- Exploiter l'inférence logique pour élaguer certains calculs numériques

Illustration

- **Réconciliation de références décrites en OWL**
  - **LN2R**: une méthode combinant inférence logique et résolution d'un système d'équations

  Thèse Paris-Sud de Fatiha Sais (novembre 2007)

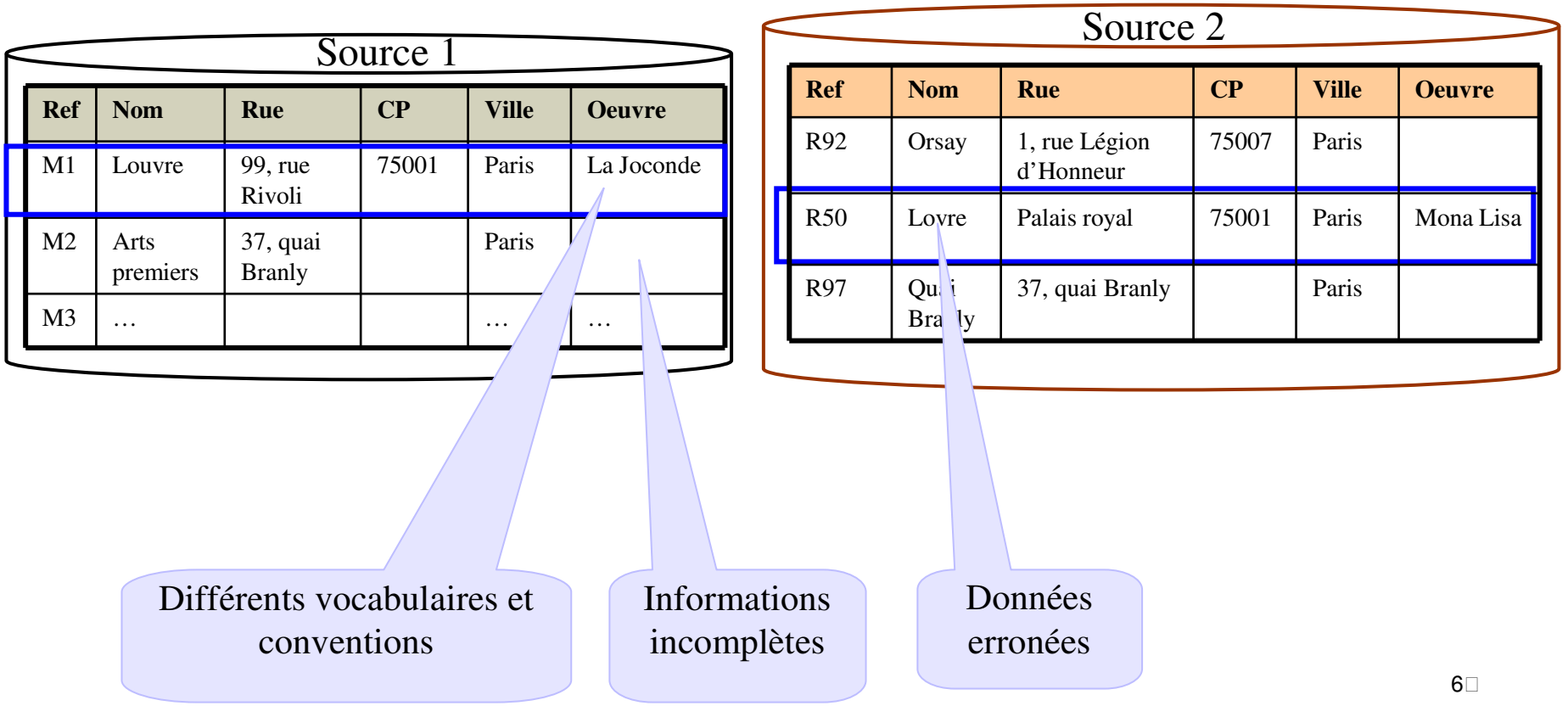
  AAAI 2007 , Journal of Data Semantics 2009
- **Alignement de taxonomies de classes**
  - **ProbaMap**: une méthode combinant logique et probabilités

  Thèse (en cours) Grenoble de Rémi Tournaire

  BDA 2009

Réconciliation de références

- Détecter que deux descriptions différentes de données représentent la même entité du monde réel.

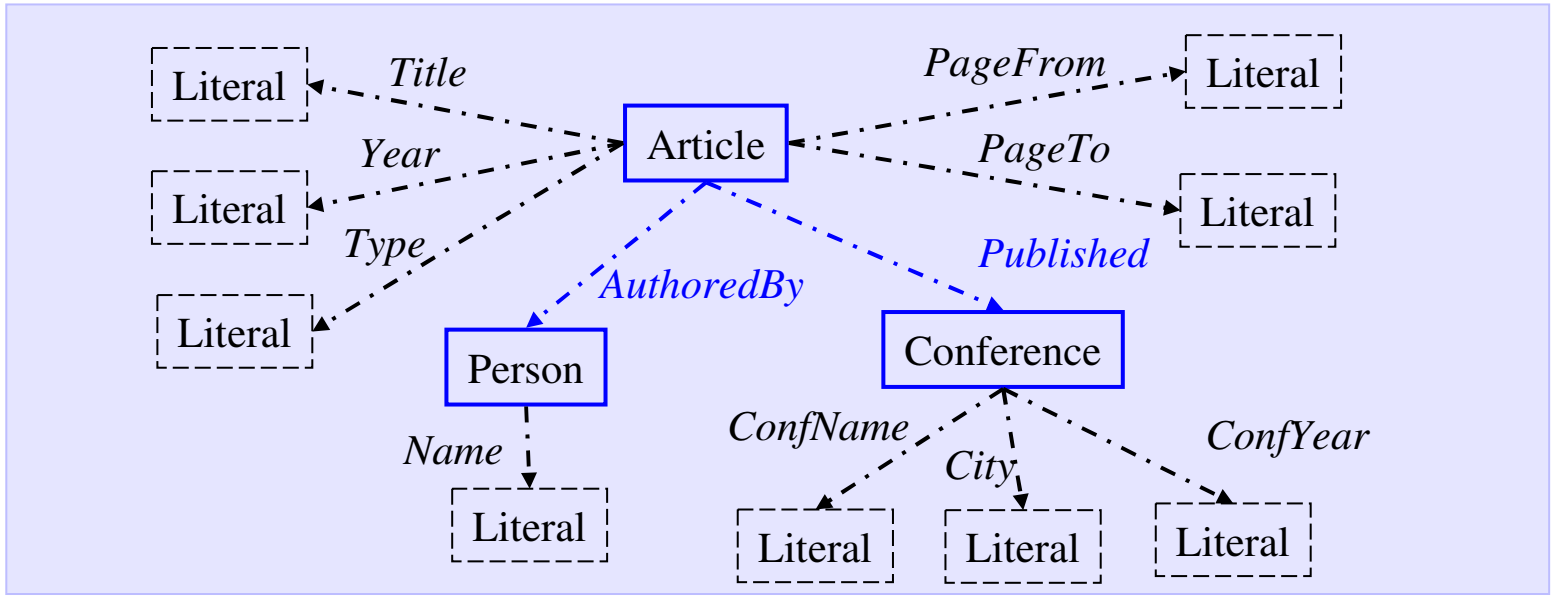


Approche LN2R

- Combinaison de deux méthodes :
  - L2R : méthode logique partielle.
  - N2R : méthode numérique qui complète les résultats de L2R.

- Approche **automatique** et **déclarative**, fondée sur la **sémantique** du schéma et des données.

Sémantique (RDFS+OWL): illustration



✓ DISJOINT(*Article,conference*), DISJOINT(*Article,Person*), DISJOINT(*Person,Conference*)

✓ Toutes les propriétés sont fonctionnelles (PF), sauf *AuthoredBy*
✓ Deux axiomes de fonctionnalité inverse combinant plusieurs attributs :

PFI(*Title,Year,Type*), PFI(*ConfName, ConfYear*)

✓ LUNA(*AuthoredBy*).

L2R: Génération automatique de règles logiques

- Traduction de UNA(src1)

  R1:src1(X) ∧ src1(Y) ∧ (X ≠ Y) ⇒ ¬Reconcile(X,Y) ; …
- Traduction de LUNA(R)

  R11(R) : R(Z, X) ∧ R(Z, Y) ∧ (X ≠ Y) ⇒ ¬Reconcile(X,Y) ; …
- Traduction de DISJOINT(C, D):

  R5(C, D) : C(X) ∧ D(Y) ⇒ ¬ Reconcile (X, Y)
- Traduction de PF(R):

  R6.1(R): Reconcile(X, Y) ∧ R(X, Z) ∧ R(Y, W) ⇒ Reconcile (Z, W)

  R6.1(Published):

  Reconcile(X,Y)∧ Published (X,Z)∧Published(Y,W) ⇒ Reconcile (Z,W)
- Traduction de PF(A):

  R6.2(A): Reconcile(X, Y) ∧ A(X, Z) ∧ A(Y, W) ⇒SynVals(Z, W)

  R6.2(ConfYear):

  Reconcile(X,Y) ∧ ConfYear(X, Z) ∧ ConfYear(Y,W) ⇒SynVals(Z,W)

L2R: algorithme d'inférence

- Application jusqu'à saturation du principe de résolution suivant la stratégie unitaire.
  - $R \cup F$: clauses de Horn sans fonctions, où :
    - R: règles mises sous forme clausale.
    - F: clauses unitaires complètement instanciées.
      - descriptions des références : faits RDF (faits-classe, faits-relation et faits-attribut).
      - faits exprimant l'origine des références : src1(i) et src2(j)
      - certains faits exprimant la synonymie ou la non synonymie de paires de valeurs: SynVals(v1, v2) ou $\neg$ SynVals(v1, v2)
  - Calcul de l'ensemble SatUnit($R \cup F$)

Propriétés de l'algorithme

- Terminaison de l'algorithme :

  assurée grâce à l'absence de symboles de fonctions.

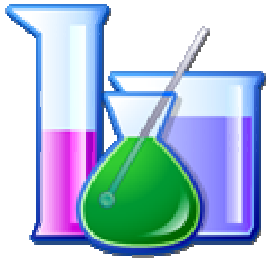
- Complétude :

  pour la déduction de toutes les clauses unitaires complètement instanciées, *Reconcile* et *SynVals*.

$\Rightarrow$ Intérêt de cette méthode logique : garantit une **précision de 100%** des résultats de réconciliation et de non réconciliation

sous réserve que les axiomes posés sont justes

Etude expérimentale du rappel de L2R



- Benchmark Cora utilisé par [Dong et al.05, Singla et Domingos'05]:
  - une collection (en RDF) de 1295 descriptions d'articles (112 articles de recherche différents), 1292 conférences et 3521 auteurs.
  - Contexte : nettoyage de données dans une source unique pour laquelle l'UNA n'est pas posée.

L2R : résultats sur Cora

	Sémantique + faits RDF	Sémantique + faits RDF + non syn. sur les dates
Rappel (REC)	52.7%	52.7%
Rappel (NREC)	50.6%	94.9%
Rappel	50.7%	94.4%

- Les résultats obtenus pour 1295 références d'Article et 1292 références de Conference.
- Pour les références de Person, nous obtenons 4298 non réconciliations en exploitant la LUNA sur la relation *AuthoredBy*.

N2R: Méthode Numérique pour la Réconciliation de Références

- Elle calcule pour chaque paire de références un score de similarité calculé sur leur description commune.

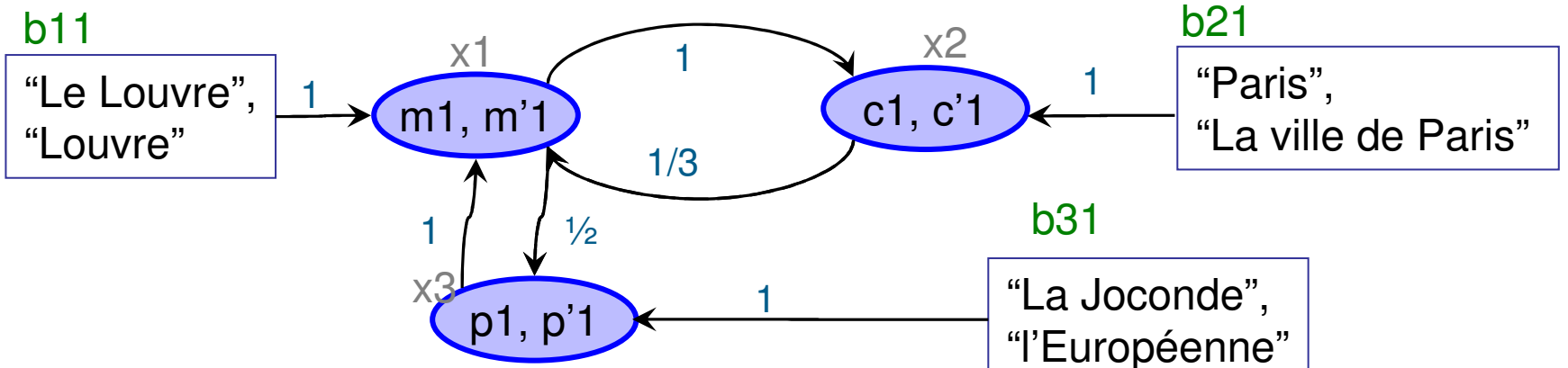
  - Utilise des algorithmes connus de calcul de similarité entre valeurs de base, e.g. Jaccard, Jaro-Winkler.
  - **Exploite aussi la sémantique**
  - Peut prendre en compte les résultats de réconciliation de la méthode logique : *Reconcile(i, i')*, ¬*Reconcile(i, i')* , *SynVals(v, v')* et ¬*SynVals(v, v')*.

Modélisation des dépendances entre similarités par un graphe

Faits RDF provenant de la source S1:
Located(m1, c1), MuseumName(m1, “le Louvre”)
Contains(m1, p1), CityName(c1, “Paris”)
PaintingName(p1, “la Joconde”)

Faits RDF provenant de la source S2 :
Located(m’1, c’1), MuseumName(m’1, “Louvre”)
Contains(m’1, p’1), CityName(c’1, “la Ville de Paris”)
PaintingName(p’1, “l’Europèenne”)

PF(Located) : (m1, m’1) influence fortement (c1, c’1)



➔ Système d’équations

Modélisation par un système d'équations

- **Variables :** similarité des paires de références.

  Une variable x_i est assignée à chaque $Sim_r(ref, ref')$

- **Equations :** expriment le calcul de similarité de chaque $Sim_r(ref, ref')$:
  - b_i : similarité des valeurs des attributs communs.
  - λ_j: pondération associée aux attributs communs et aux relations communes de x_i.

Système d'équations non linéaires

$$x_i = \max\left(\max\left(\bigcup_{j=0}^{j=|DF_A(<ref,ref'>)|} (b_{ij-df}), \bigcup_{j=0}^{j=|DF_R(<ref,ref'>)|} (x_{ij-df}), \right), \right.$$

$$\left(\sum_{j=0}^{j=|NDF_A(<ref,ref'>)|} (\lambda_{ij} * b_{ij-ndf}) + \sum_{j=0}^{j=|NDF_A^*(<ref,ref'>)|} (\lambda_{ij} * BS_{ij-ndf}) + \right.$$

$$\left. \left. \sum_{j=0}^{j=|NDF_R(<ref,ref'>)|} (\lambda_{ij} * x_{ij-ndf}) + \sum_{j=0}^{j=|NDF_R^*(<ref,ref'>)|} (\lambda_{ij} * XS_{ij-ndf}) \right) \right)$$

- NDF(x_i), calculé par une moyenne pondérée
- DF(x_i), calculé par un maximum

➔ Système d'équations non linéaires

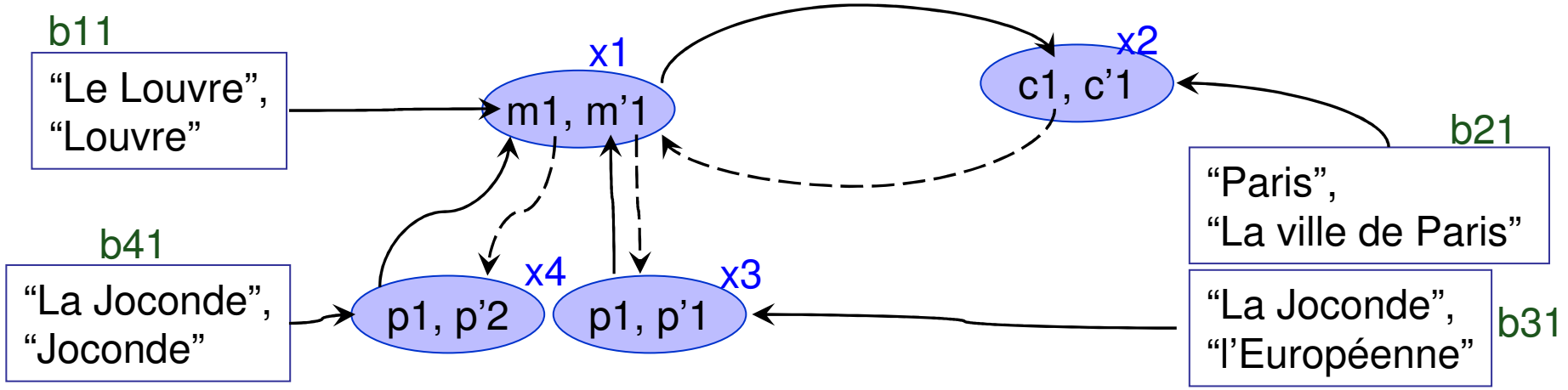
Résolution du système d'équations non linéaires

- Une méthode itérative inspirée de la méthode de *Jacobi.*
  - Initialisation des valeurs des variables x_i à 0.
  - Affinement itératif des valeurs de chaque x_i en utilisant les valeurs des x_i calculées à l'itération précédente.
  - Terminaison: un point fixe avec une précision ε

$$\forall\, x_i \qquad |x_i^k - x_i^{k-1}| < \varepsilon$$

➔Preuve de convergence.

Illustration



x1 = **max**(b11, x3, x4, 1/4 * x2)

x2 = **max**(b21, x1)

x3 = **max**(b31, 1/2* x1)

x4 = **max**(b41 , 1/2 * x1)

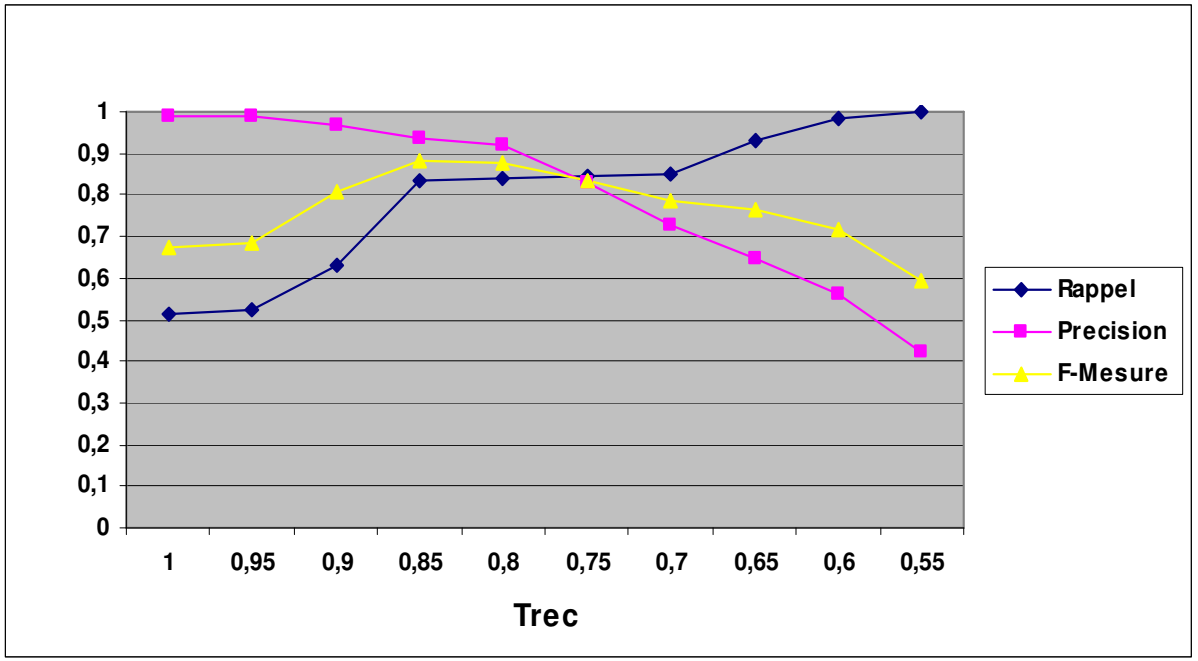
	x1	x2	x3	x4
Initialisation	0.0	0.0	0.0	0.0
Itération 1	0.8	0.3	0.1	0.7
Itération 2	0.8	0.8	0.4	0.7
Itération 3	0.8	0.8	0.4	0.7

λ= 1/(| CAttr | + | CRel |) ε = 0.02

b11 = 0.8, b21 = 0.3, b31 = 0.1, b41 = 0.7

Solution : x1 = 0.8
x2 = 0.8
x3 = 0.4
x4 = 0.7

N2R : les résultats sur Cora



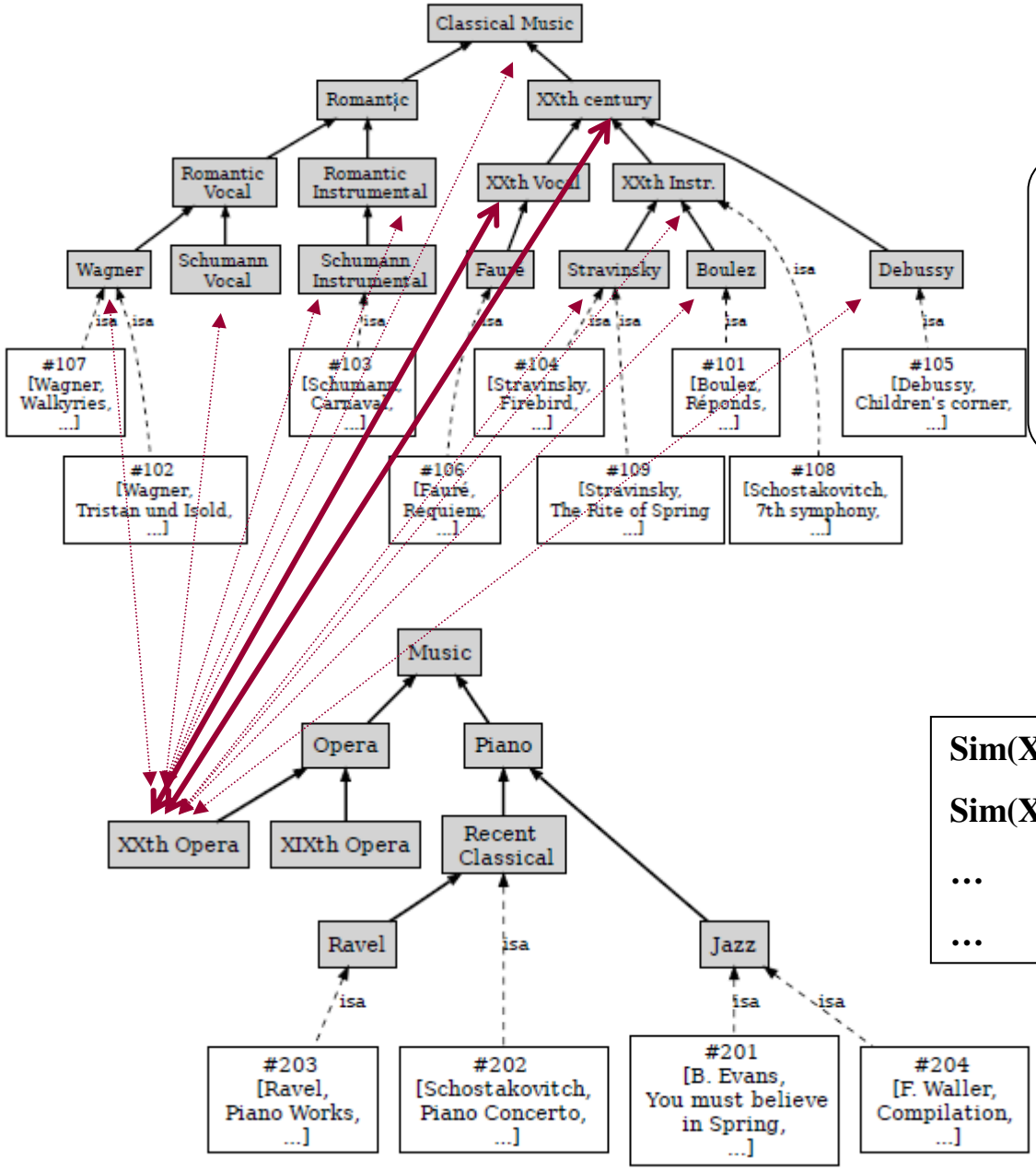
- Trec=1, les réconciliations obtenues par L2R sont aussi obtenues par N2R.
- Trec=1 à Trec=0.85, le rappel croît de 33 % alors que la précision décroît seulement de 6 %.
- Trec = 0.85, la F-mesure est de 88 % :
  - Meilleurs que ceux obtenus par la méthode supervisée de [Singla et Domingos'05]
  - Inférieurs à ceux obtenus (97 %) par la méthode supervisée de [Dong et al.'05]

Combiner logique et probabilités pour l'alignement d'ontologies

Alignement d'ontologies

- Un problème très difficile au cœur du Web sémantique
- Un domaine de recherche très actif
  - Une compétition internationale annuelle
  - Des méthodes nombreuses
    - calcul de similarité entre classes ou propriétés
    - difficile d'interpréter les résultats

Illustration



- quels mappings garder ?
- avec quelle sémantique ?

Calcul de similarités

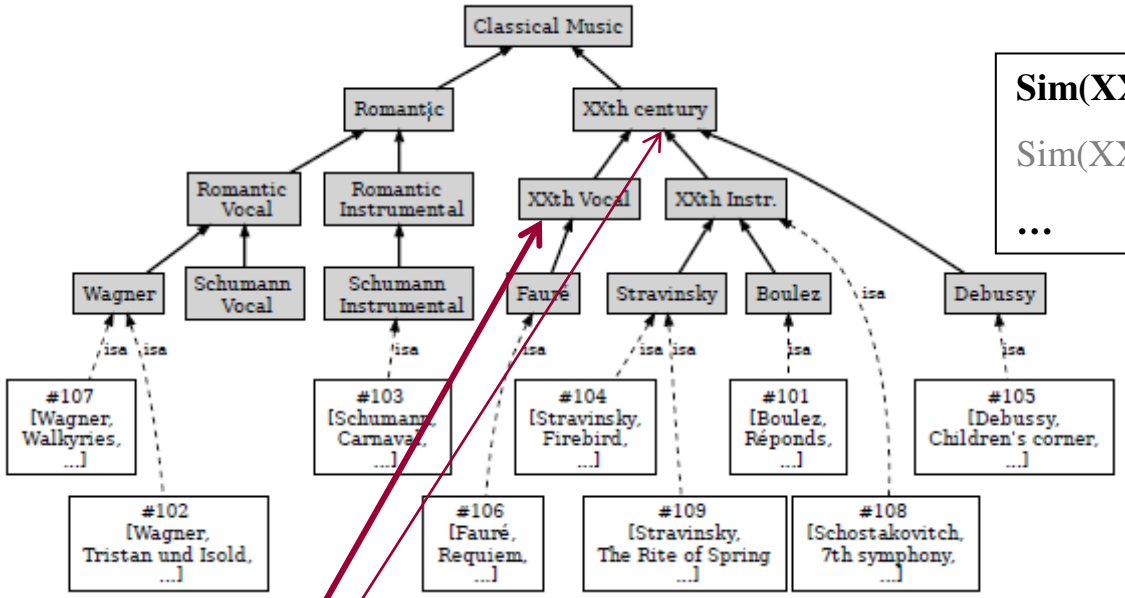
Sim(XXth Opera, XXth Vocal) = 0.78

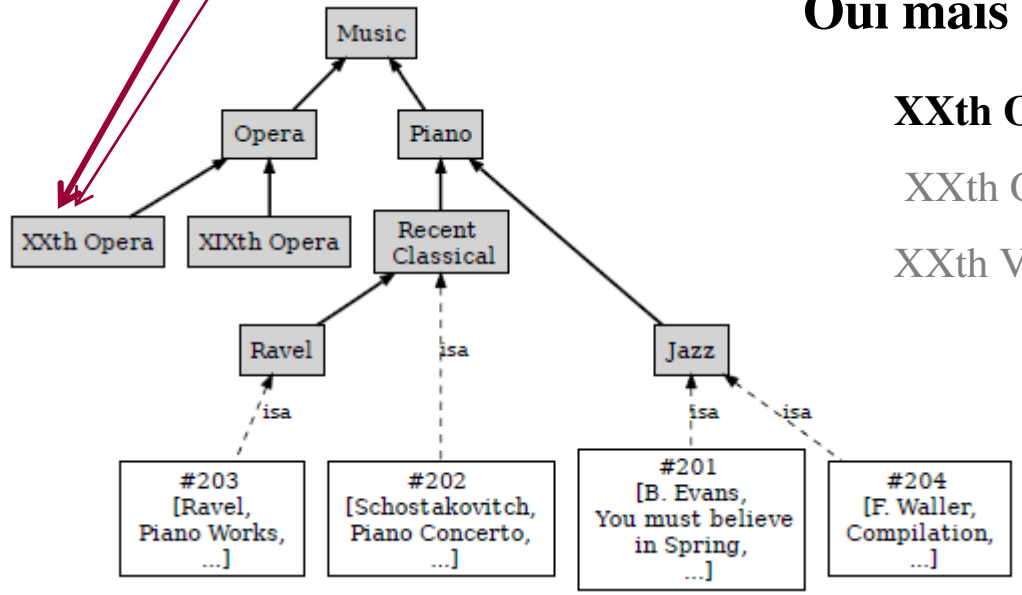
Sim(XXth Opera, XXth century) = 0.74

…

…

les « bons » mappings sont ceux au dessus d'un seuil





Similarités

Sim(XXth Opera, XXth Vocal) = 0.78

Sim(XXth Opera, XXth century) = 0.74

…

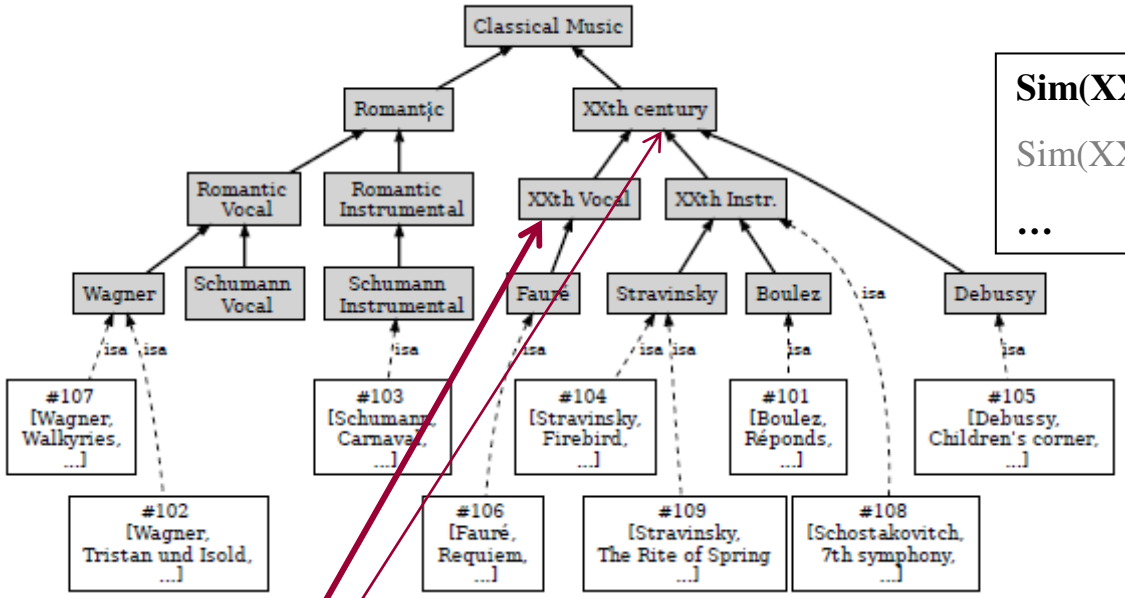
Oui mais … quelle sémantique ?

XXth Opera ≡ XXth Vocal

XXth Opera $\sqsubseteq$ XXth Vocal

XXth Vocal $\sqsubseteq$ XXth Opera

les « bons » mappings sont ceux au dessus d'un seuil

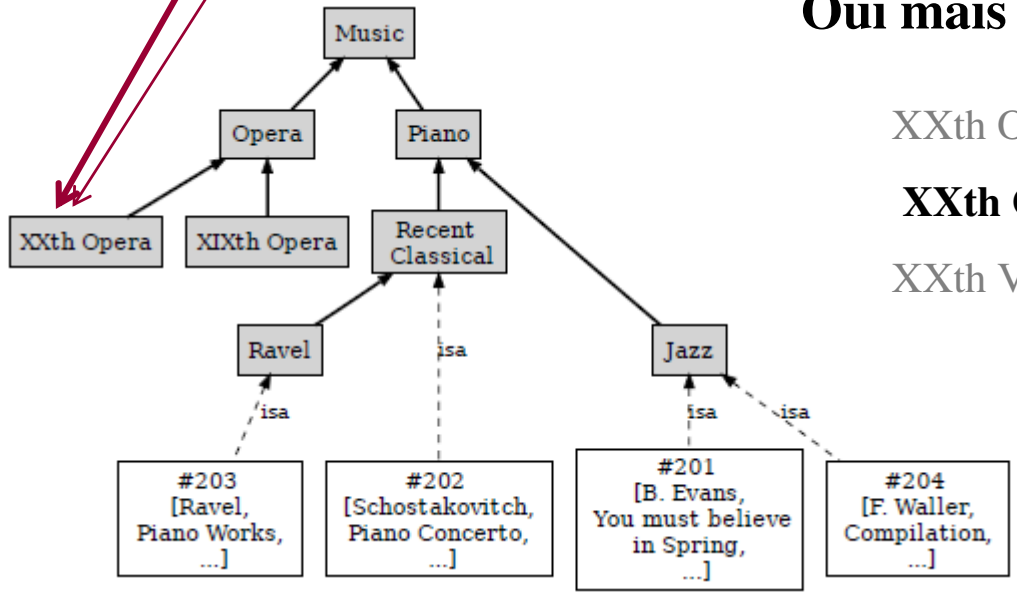


similarities

Sim(XXth Opera, XXth Vocal) = 0.78

Sim(XXth Opera, XXth century) = 0.74

…



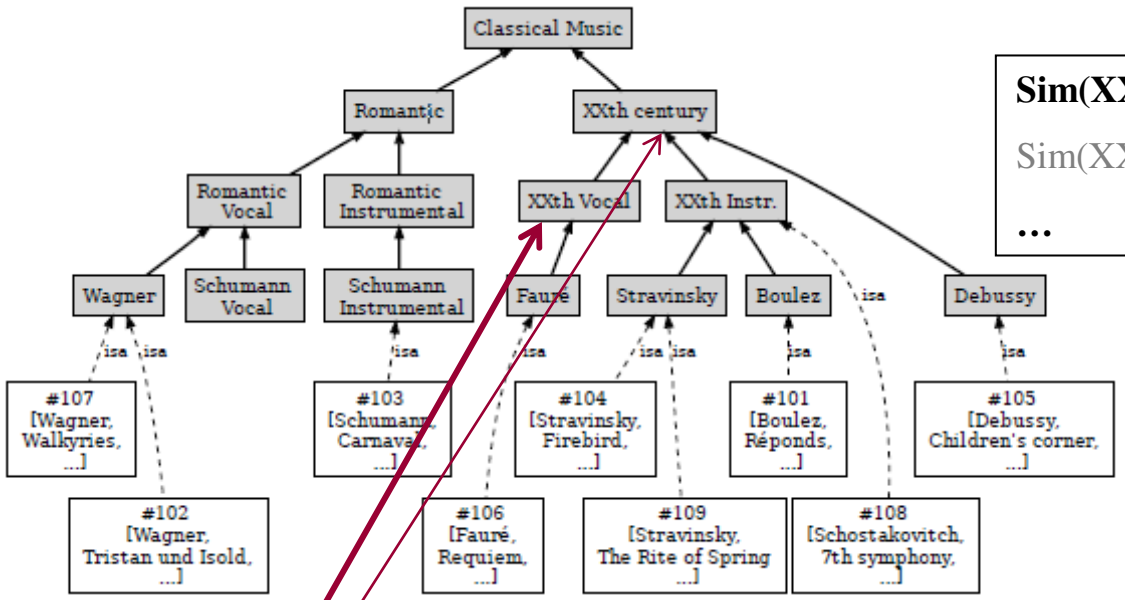
Oui mais … quelle sémantique ?

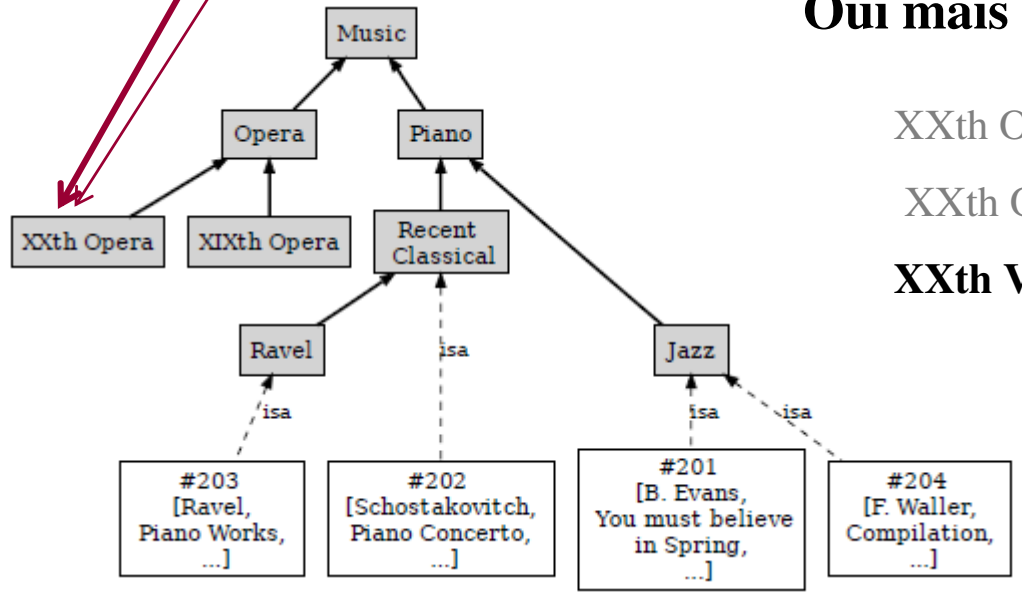
XXth Opera ≡ XXth Vocal

XXth Opera ⊑ XXth Vocal

XXth Vocal ⊑ XXth Opera

les « bons » mappings sont ceux au dessus d'un seuil





similarities

Sim(XXth Opera, XXth Vocal) = 0.78

Sim(XXth Opera, XXth century) = 0.74

…

Oui mais … quelle sémantique ?

XXth Opera ≡ XXth Vocal

XXth Opera $\sqsubseteq$ XXth Vocal

XXth Vocal $\sqsubseteq$ XXth Opera

Pourquoi donner une sémantique logique aux mappings ?

- Pour les utiliser !
  - Reformuler et/ou enrichir des requêtes
  - Trouver des réponses bien fondées

XXth Opera ≡ XXth Vocal

XXth Opera ⊑ XXth Vocal

XXth Vocal ⊑ XXth Opera

Les 2 classes sont synonymes:

• elles correspondent à des requêtes ou à des annotations équivalentes

• on peut obtenir des réponses à une requête portant sur l'une en renvoyant des ressources annotées par l'autre

Pourquoi donner une sémantique logique aux mappings ?

- Pour les utiliser !
  - Reformuler et/ou enrichir des requêtes
  - Trouver des bonnes réponses

XXth Opera ≡ XXth Vocal

XXth Opera $\sqsubseteq$ **XXth Vocal**

XXth Vocal $\sqsubseteq$ XXth Opera

La classe **XXth Vocal** est plus générale que la classe **XXth Opera**

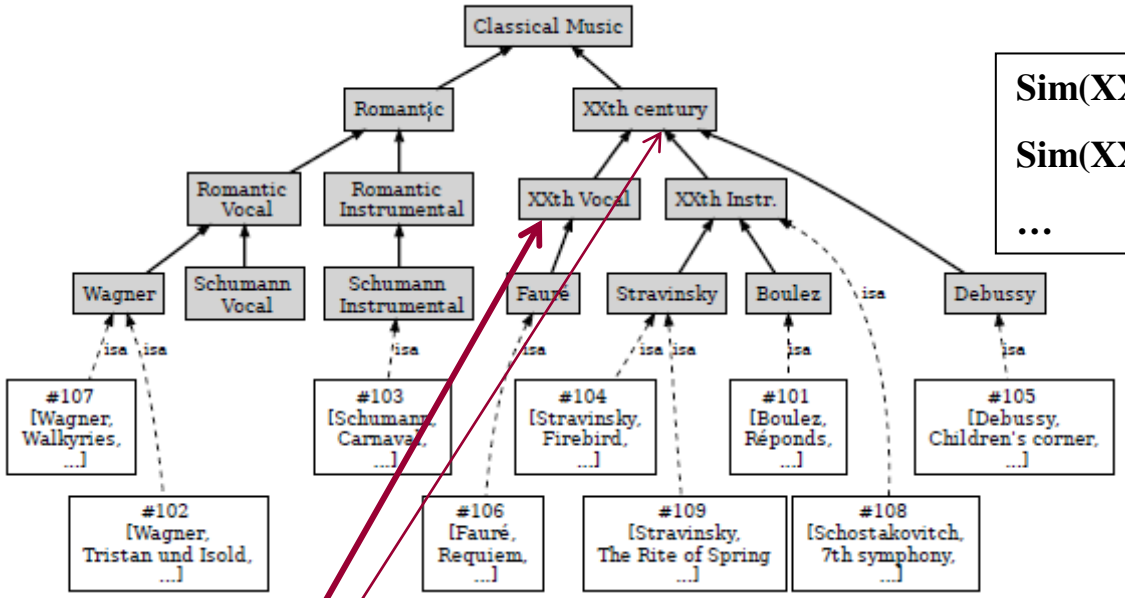
• on peut obtenir des réponses à une requête sur **XXth Vocal** en renvoyant des ressources annotées par **XXth Opera**

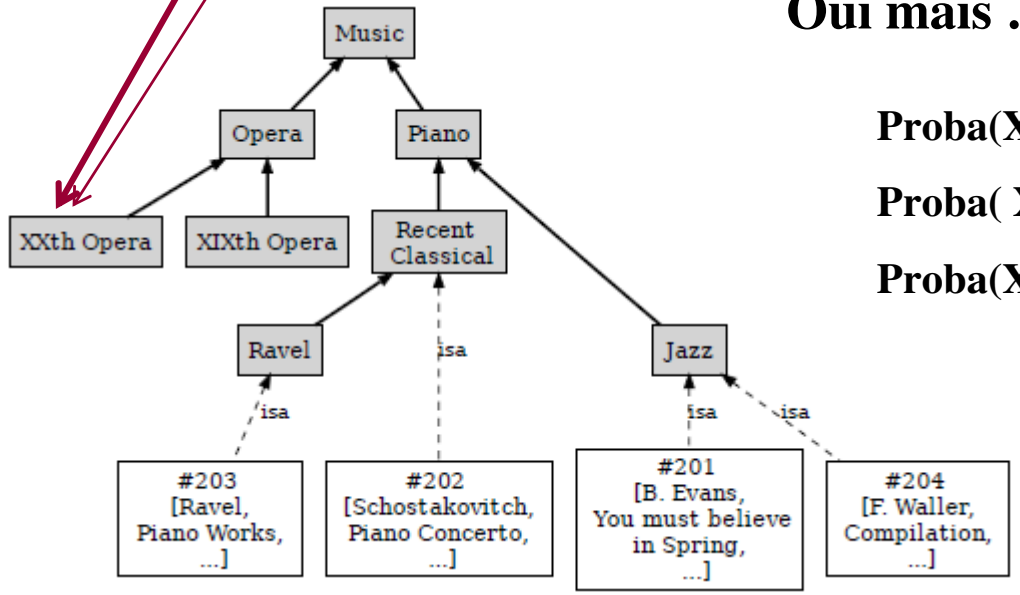
- **être instance de XXth Opera implique logiquement être instance de XXth Vocal**

• mais l'inverse ne fournit pas de réponses certaines à la requête

- **#106 [Fauré, Requiem]**

Interpréter les similarités comme des probabilités





Similarités

Sim(XXth Opera, XXth Vocal) = 0.78

Sim(XXth Opera, XXth century) = 0.74

…

Oui mais … quelle sémantique ?

Proba(XXth Opera ≡ XXth Vocal) = 0.78

Proba(XXth Opera ⊑ XXth Vocal) = ??

Proba(XXth Vocal ⊑ XXth Opera) = ??

Pourquoi donner une sémantique probabiliste aux mappings ?

- Pour les utiliser ! … de manière plus flexible que si on les interprétait de manière logique mais de manière fondée
  - Pour inférer la probabilité que **#106 [Fauré, Requiem]** soit une réponse à la requête **XXth Opera** à partir de :
    - **la probabilité du mapping entre XXth Opera**
    - **le fait que #106 [Fauré, Requiem] est instance de XXth Vocal**
- Difficulté :
  - interférence entre sémantique probabiliste des mappings et sémantique logique des relations entre classes de chaque taxonomie:

    Puisque **XXth Vocal** $\sqsubseteq$ **XXth century** on devrait avoir **:**

    Proba(XXth Opera $\sqsubseteq$ **XXth Vocal)** $\leq$ **Proba (XXth Opera** $\sqsubseteq$ **XXth century)**
  - Les similarités ne garantissent pas cette propriété

    Sim(XXth Opera, XXth Vocal) = 0.78 **Sim(XXth Opera, XXth century) = 0.74**

Notre approche dans ProbaMap

- Comparaison de deux sémantiques probabilistes compatibles avec la sémantique logique
  - Probabilité conditionnelle $P_c(E_1 \sqsubseteq F_2) = P(F_2|E_1)$
  - Probabilité ensembliste $P_u(E_1 \sqsubseteq F_2) = P(\overline{E_1} \cup F_2)$

$$P(\overline{E_1} \cup F_2) = 1 - P(E_1 \setminus F_2)$$

$$P(F_2|E_1) = 1 - \frac{P(E_1 \setminus F_2)}{P(E_1)}$$

$$P(F_2|E_1) \leq P(\overline{E_1} \cup F_2)$$

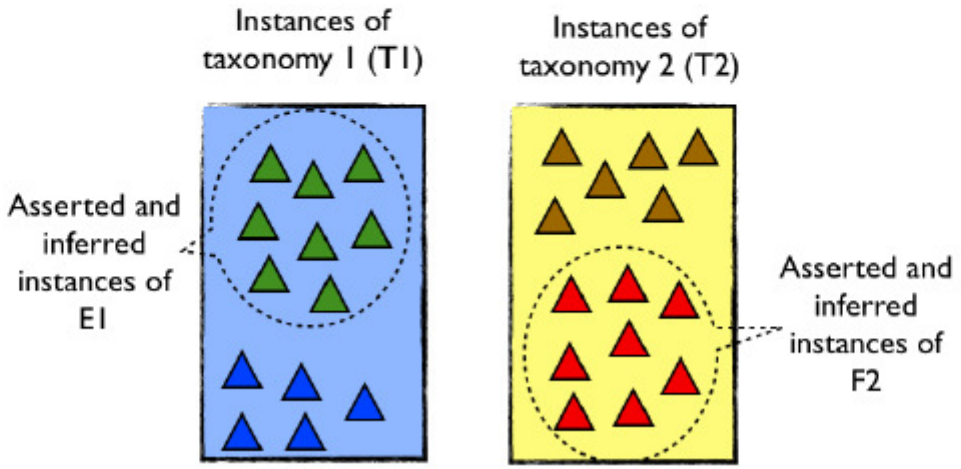
- Estimation Bayésienne de ces probabilités
- Un algorithme de découverte de mappings les plus probables entre taxonomies
  - combinant les deux probabilités pour plus de robustesse

Estimation Bayésienne des probabilités

$$P_u, P_c = f(P(E_1), P(E_1 \cap F_2))$$

Bayesian estimation for $P(E_1)$ and $P(E_1 \cap F_2)$ [DeGroot, 2004]

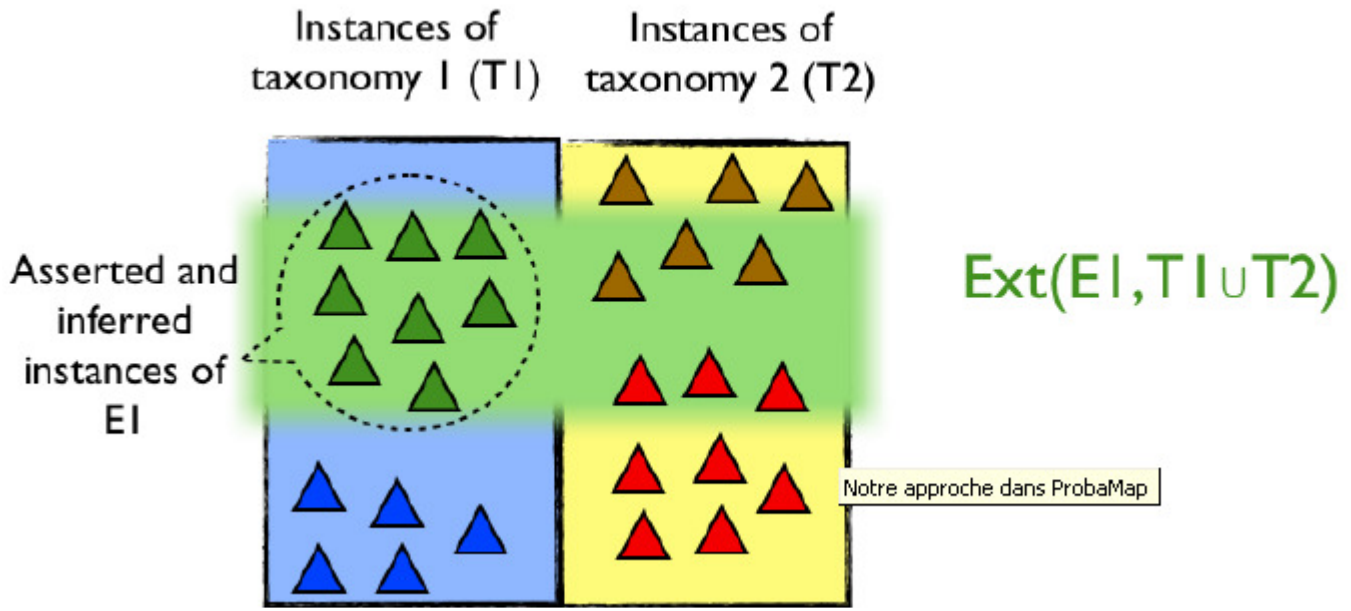
$$P(E_1) \approx \frac{1+|Ext(E_1, T_1 \cup T_2)|}{2+|Ext(T_1 \cup T_2)|} \qquad P(E_1 \cap F_2) \approx \frac{1+|Ext(E_1 \cap F_2, T_1 \cup T_2)|}{4+|Ext(T_1 \cup T_2)|}$$



Problème : comment calculer l'extension de l'intersection des classes E1 et F2 quand elles sont peuplées indépendamment ?

Par classification

- En utilisant des classifieurs
  - entrainés sur les méta-données ou le contenu des fichiers déclarés ou inférés logiquement comme instances des classes E1 et F2 dans chaque taxonomie
  - pour prédire quelles instances de F2 doivent être classées dans E1 et vice versa



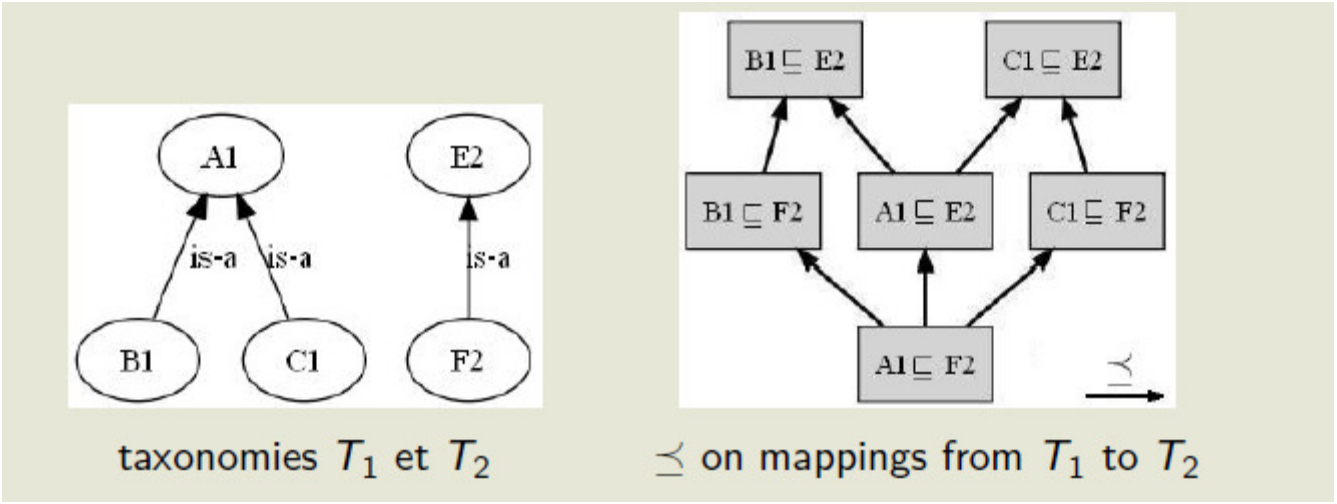
L'algorithme ProbaMap

- Input :
  - 2 taxonomies T1 et T2 + leurs instances (classement et description)
  - 2 seuils
- Output : l'ensemble des mappings dont la probabilité dépasse un certain seuil
  - seuils = 0 : les probabilités de tous les mappings
- 3 variantes selon les 3 critères suivants de validité de mappings

1. $P_c(m) \geq S_c$
2. $P_u(m) \geq S_u$
3. $P_c(m) \geq S_c$ et $P_u(m) \geq S_u$

Le principe

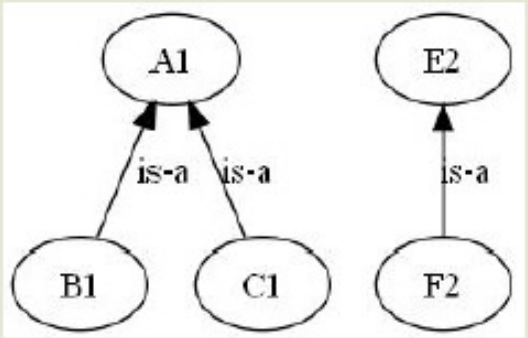
- Énumération et évaluation de la probabilité des mappings selon un ordre logique induit de l'ordre logique entre classes dans chaque taxonomie



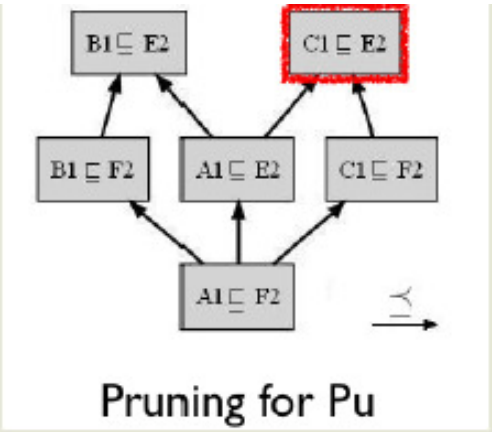
- Elagage de l'étape d'évaluation exploitant une propriété de monotonie des probabilités par rapport à l'ordre logique

- $m \preceq m' \Rightarrow P_u(m) \leq P_u(m')$
- $m \preceq m'$ **and** $lhs(m) = lhs(m') \Rightarrow P_c(m) \leq P_c(m')$

Elagage



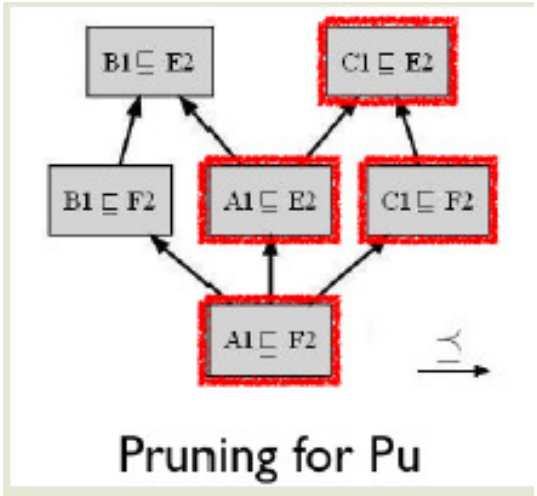
taxonomies T_1 et T_2

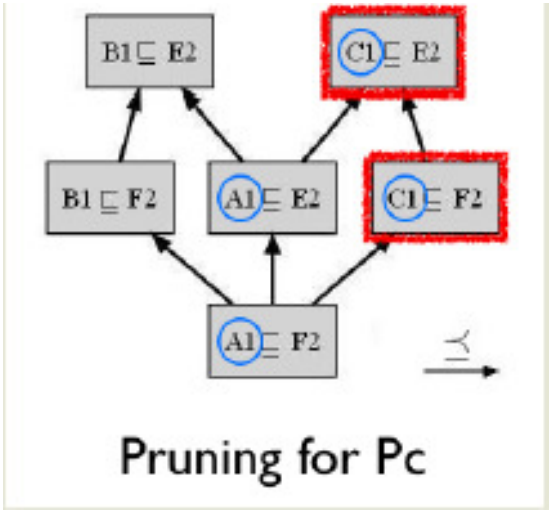


$\preceq$ on mappings from T_1 to T_2

Pruning exploiting monotonicity :

if $P_u(m) < s$, for all m' implicants of m ($m' \preceq m$), $P_u(m') < s$





Expérimentations

- Sur données réelles et synthétiques
  - L'élagage est plus important avec Pc qu'avec Pu (encore un peu amélioré quand on combine les deux)
  - La précision est meilleure quand la validité des mappings dépend de Pc
  - La classification utilisant C4.5 ou SVM mène à une précision et un rappel bien meilleurs qu'en utilisant Naive Bayes
  - Très bon passage à l'échelle (grâce à l'élagage)
    - Test sur deux taxonomies benchmarks de OAEI de 2600 et 6000 classes
      - de l'ordre de 30 millions de mappings possibles

Conclusion

- De plus en plus de données et de connaissances accessibles via le Web
  - Le Web (sémantique) devient une immense base de connaissances constituée d'ilôts pré-existants entre lesquels on souhaite jeter des ponts
  - Ces ponts sont des correspondances (entre données ou connaissances) qui sont par nature incertaines
- Des similarités ne sont pas nécessairement interprétables comme des probabilités
- L'intérêt de découvrir des probabilités permet de greffer un raisonnement probabiliste

MERCI …