

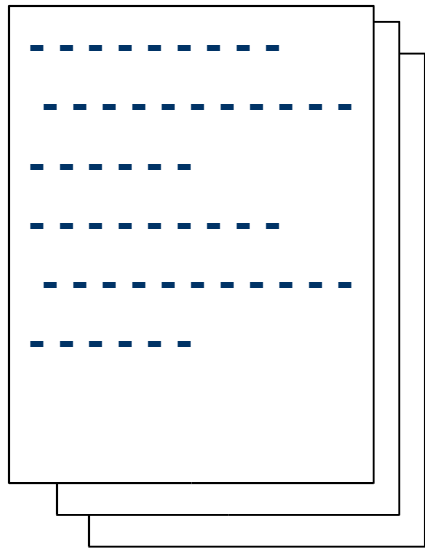
Classification conceptuelle

Mathieu Roche

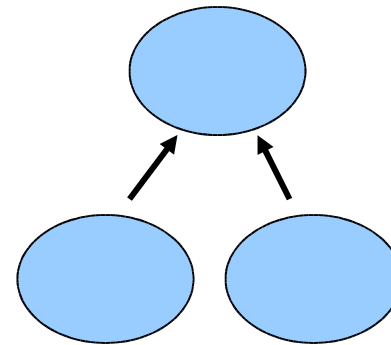
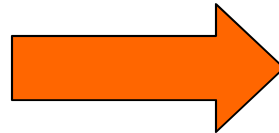
Cours Fouille de Données

février 2007

Construction des classes



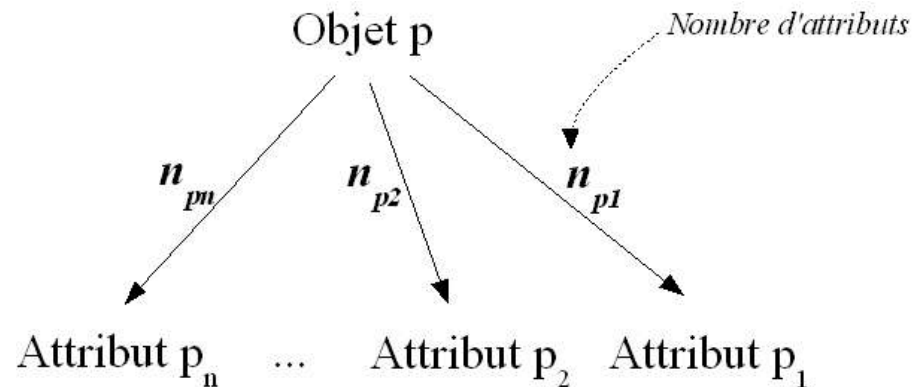
Corpus brut



***Classification
conceptuelle***

Asium (1/9)

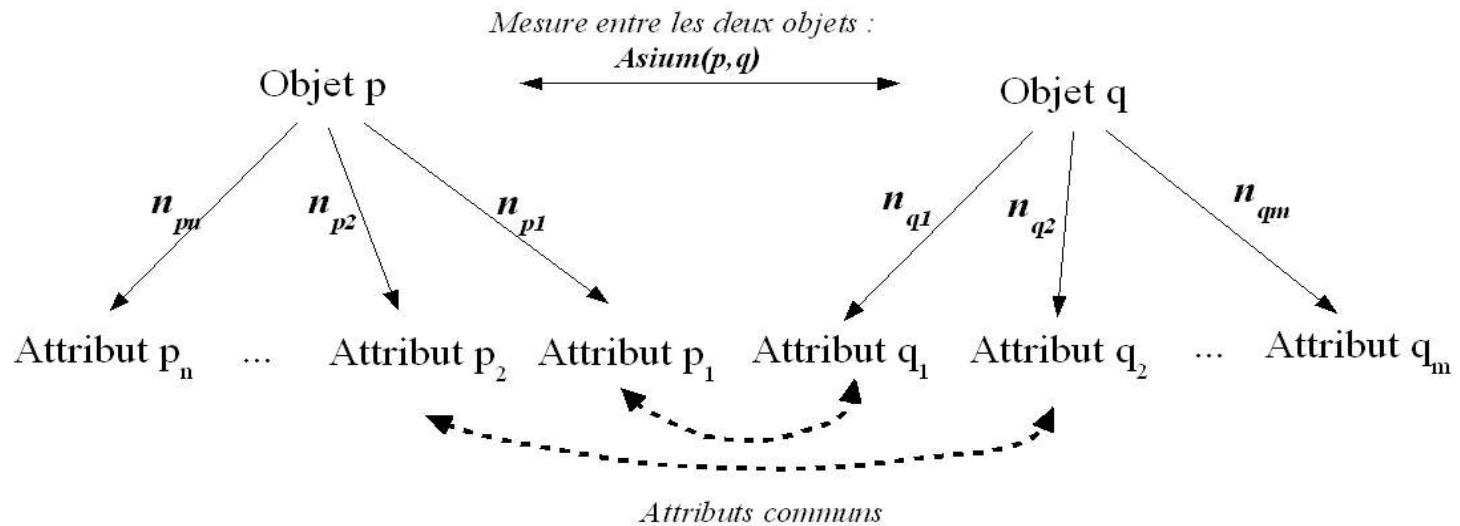
- *Asium* [Faure et Nedellec, 1998] utilise en entrée les textes d'un domaine analysés syntaxiquement. Il va ensuite extraire les triplets:
 - verbe,
 - préposition/fonction (si pas de préposition),
 - nom de tête du complément en forme lémmatisée (Attribut).



Asium (2/9)

- Puis, on rassemble tous les noms apparaissant après un couple verbe/préposition (ou fonction). Ces listes de noms sont appelées classes de base. Elles sont reliées aux couples (verbe/préposition, fonction) qui ont permis de les créer.
- *Asium* calcule ensuite une similarité entre toutes ces classes de base deux à deux. Les plus proches vont être assemblées pour former les classes apprises.
- Ces classes apprises représentent les **concepts du domaine**.

Asium (3/9)



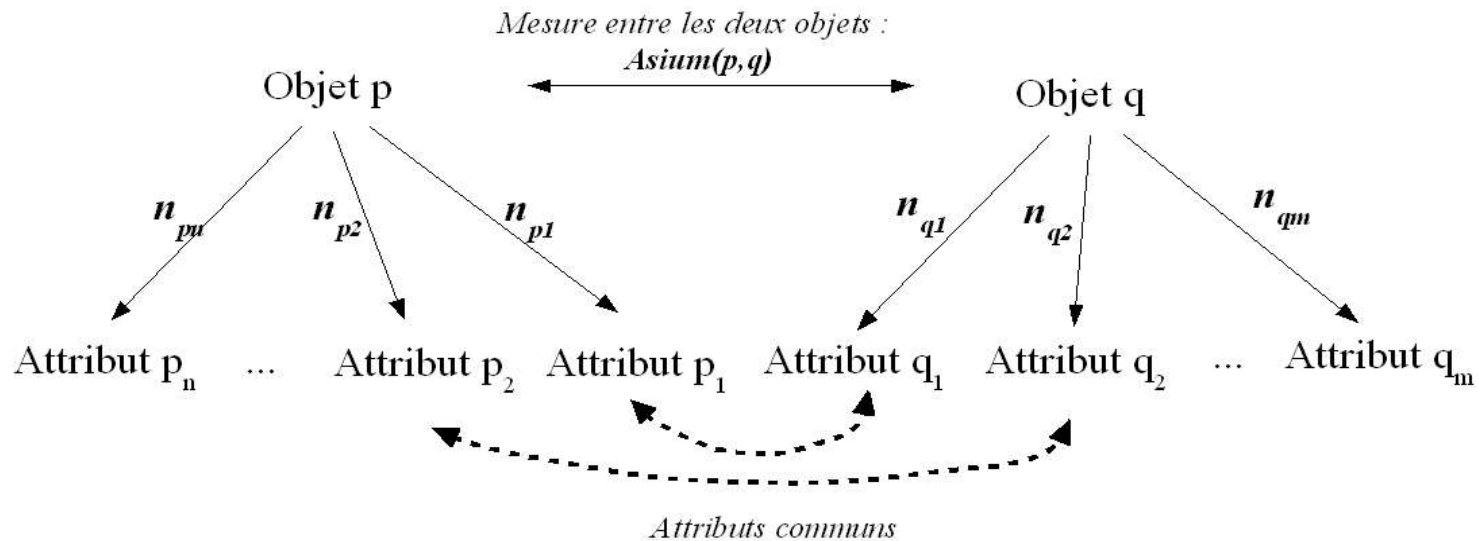
- Nous donnons la mesure d'Asium (notée $Asium$) entre deux objets p et q ayant respectivement comme attributs p_1, \dots, p_n et q_1, \dots, q_m . Plus $Asium(p, q)$ est proche de 1 et plus les objets p et q sont proches d'un point de vue sémantique.

Asium (4/9)

$$Asium(p, q) = \frac{\log_{Asium}(\sum NbOccCom_q(p_i)) + \log_{Asium}(\sum NbOccCom_p(q_i))}{\log_{Asium}(\sum NbOcc(p_i)) + \log_{Asium}(\sum NbOcc(q_i))}$$

- $NbOccCom_q(p_i)$ représente le nombre d'occurrences des attributs p_i en relation avec l'objet p qui sont aussi des attributs de l'objet q .
- $NbOcc(p_i)$ représente le nombre d'occurrences des attributs p_i .
- Enfin \log_{Asium} est égale à la fonction \log à un décalage près afin d'éviter les problèmes de calculs dans le cas où une somme est égale à zéro :
 - Si $x = 0$ alors $\log_{Asium}(x) = 0$
 - Sinon $\log_{Asium}(x) = \log(x) + 1$

Asium (5/9)



$$Asium(p, q) = \frac{\log_{Asium}(n_{p1} + n_{p2}) + \log_{Asium}(n_{q1} + n_{q2})}{\log_{Asium}(n_{p1} + n_{p2} + n_{p3} + \dots + n_{pn}) + \log_{Asium}(n_{q1} + n_{q2} + n_{q3} + \dots + n_{qm})}$$

Asium (6/9)

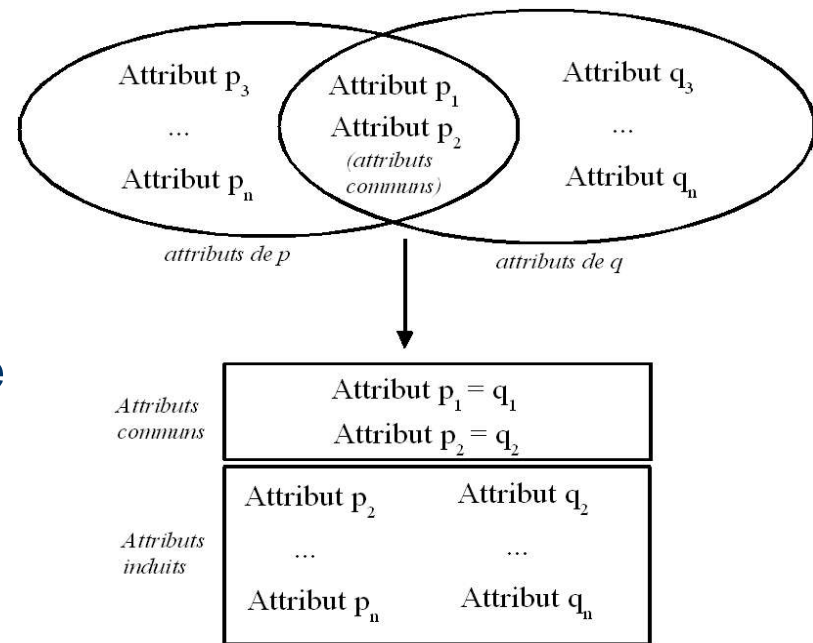
Action « appuyer_sur »	Action « relâcher_COD »
<appuyer_sur, touche, 10> <appuyer_sur, barre, 2>	<relâcher_COD, touche, 4> <relâcher_COD, bouton, 3>

- Le but est de calculer la mesure entre les deux actions « *appuyer_sur* » et « *relâcher_COD* » qui possèdent l'attribut commun *touche*. Dans cet exemple nous avons donc :

$$Asium(\text{appuyer sur}, \text{relâcher COD}) = \frac{\log_{Asium}(10) + \log_{Asium}(4)}{\log_{Asium}(10+2) + \log_{Asium}(4+3)} = 0,92$$

Asium (7/9)

- Les objets les plus proches étant déterminés, tous les attributs de ces deux objets sont rassemblés dans une même classe sémantique. Cette classe est composée des attributs communs et de la réunion des complémentaires qui sont appelés les attributs induits ou inférés.



Exemple : { **touche** (*attribut commun*), **barre** (*attribut induit*), **bouton** (*attribut induit*) }

Asium (8/9)

- Le simple calcul de similarité n'est pas suffisant pour apprendre les concepts d'un domaine, **l'aide d'un expert est primordiale**. En effet, certaines classes apprises peuvent comporter du bruit (erreurs d'analyse syntaxique).
- Par exemple, les deux classes de base suivantes:
 - **C1**: *voyager en (bateau, été, avion, hiver, voiture, train)*
 - **C2**: *se déplacer en (bateau, hiver, 4x4, vélo, avion)*

ont une bonne similarité. Néanmoins, leur agrégation ne représente pas un mais **deux concepts**. L'expert interviendra donc pour découper la classe apprise en deux concepts:

Moyens de transport et Saisons.

Asium (9/9)

- De plus, l'expert devra vérifier que les inductions effectuées par *Asium* sont correctes. Ici les inductions effectuées sont :
 - *voyager en 4x4*
 - *voyager en vélo*
 - *se déplacer en été*
 - *se déplacer en voiture*
 - *se déplacer en train*
- Ces utilisations n'étant pas présentes dans les textes mais **découvertes** par *Asium* (induction).
- Le calcul de similarité s'effectue entre toutes les classes de base deux à deux, puis l'expert valide la liste de toutes les classes apprises par *Asium*.

LEXICLASS (1/5)

- Entrées de LEXICLASS (Assadi, 1998) : termes extraits avec LEXTER et/ou SYNTEX
- Avec LEXTER et SYNTEX chaque groupe nominal est décomposé en tête et expansion à l'aide de règles grammaticales.
- Construction d'une table individus/variables.

LEXICLASS (2/5)

- Exemple : termes Nom-Adjectif (corpus de Cvs en français)
- individus -> noms / variables -> adjectifs.

	commercial	informatique	mécanique	électronique
BEP	1	1	1	1
CAP	1	0	1	1
BTS	1	1	0	0

LEXICLASS (3/5)

- mesure de similarité pour chaque couple de mots (par exemple, entre les noms « BEP » et « CAP » puis entre « BEP » et « BTS », etc.)
- Utilisation de la mesure de Jaccard (adaptées aux données binaires) :

$$Jaccard(A, B) = \frac{A \cap B}{A \cup B}$$

LEXICLASS (4/5)

- $a(x,y)$ = nombre de mots (variables) présents à la fois chez x et chez y
- $b(x,y)$ = nombre de mots (variables) présents chez x et absents chez y
- $c(x,y)$ = *nombre de mots (variables) présents chez y et absents chez x*

- Mesure de Jaccard :

$$Jaccard(x, y) = \frac{a(x, y)}{a(x, y) + b(x, y) + c(x, y)}$$

LEXICLASS (5/5)

- Exemple :

$$Jaccard(BEP, CAP) = \frac{3}{3+1+0} = 0.75$$

	commercial	informatique	mécanique	électronique
BEP	1	1	1	1
CAP	1	0	1	1
BTS	1	1	0	0

- Lors de l'étape suivante, le nouvel objet constitué des mots CAP et BEP peut être rapproché du nom « BTS ».
- Construction d'un arbre de classification hiérarchique qui met en évidence l'inclusion progressive des classes formées.

