Levelized Taxonomy Approach for the Job Seeking/Recruitment Problem

Michaël Guedj

UTEC INT, School of IT and New Technologies
UTEC – CCI Seine-et-Marne
michael.guedj@seineetmarne.cci.fr

CSE 2016 – Paris, France

Presentation Outline

Presentation Outline

The Job Seeking/Recruitment Problem

Goal:

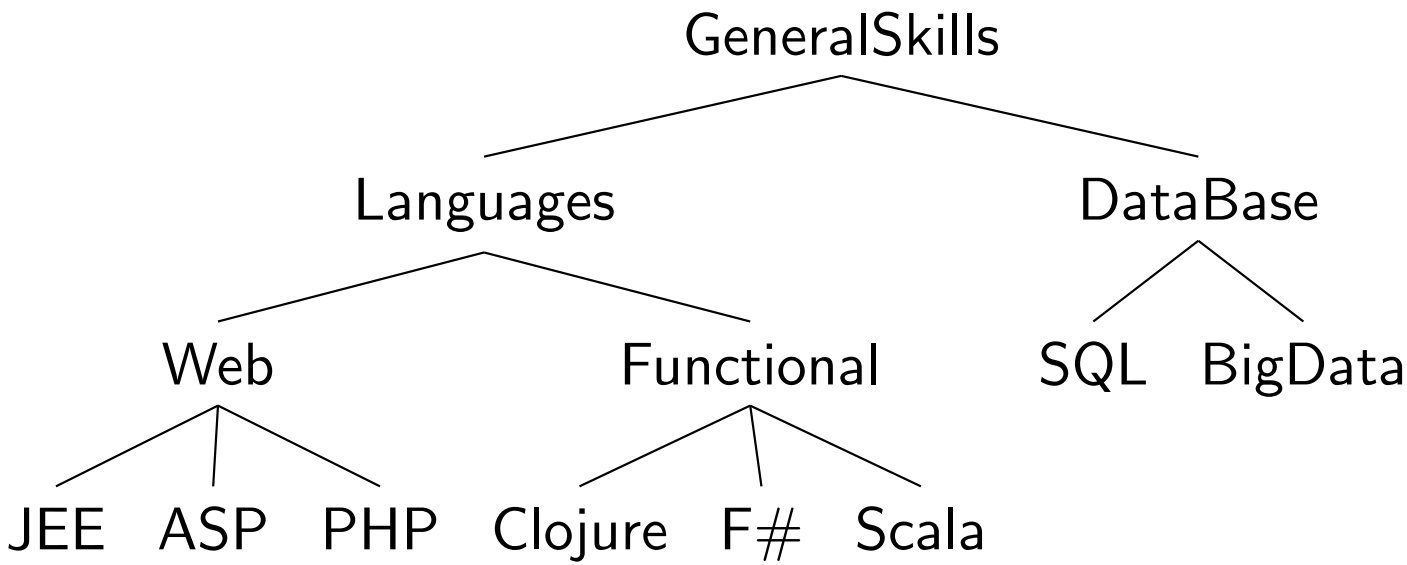
- Retrieve a list of job positions to a job applicant based on his/her preferences;
- Generate a list of job candidates to a recruiter based on the job requirements.

Presentation Outline

Resolution [Zhong-al 02] and [Bizer-al 05]

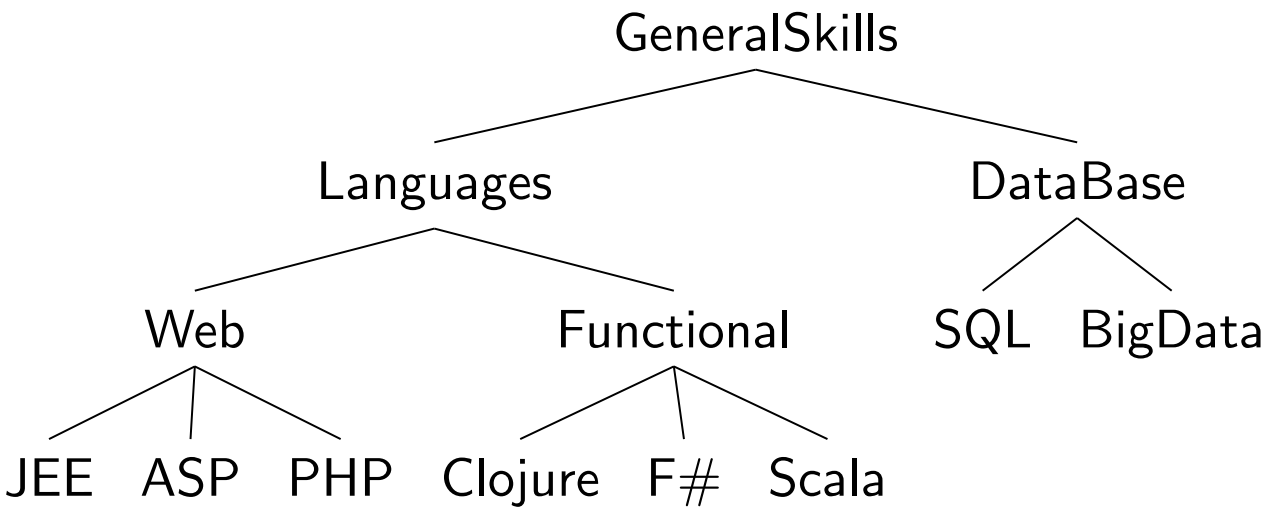
- Profiles of the actors using concepts from a taxonomy.
- Semantic Ranking using this taxonomy.

Example of a Taxononomy (Skills Hierarchy)



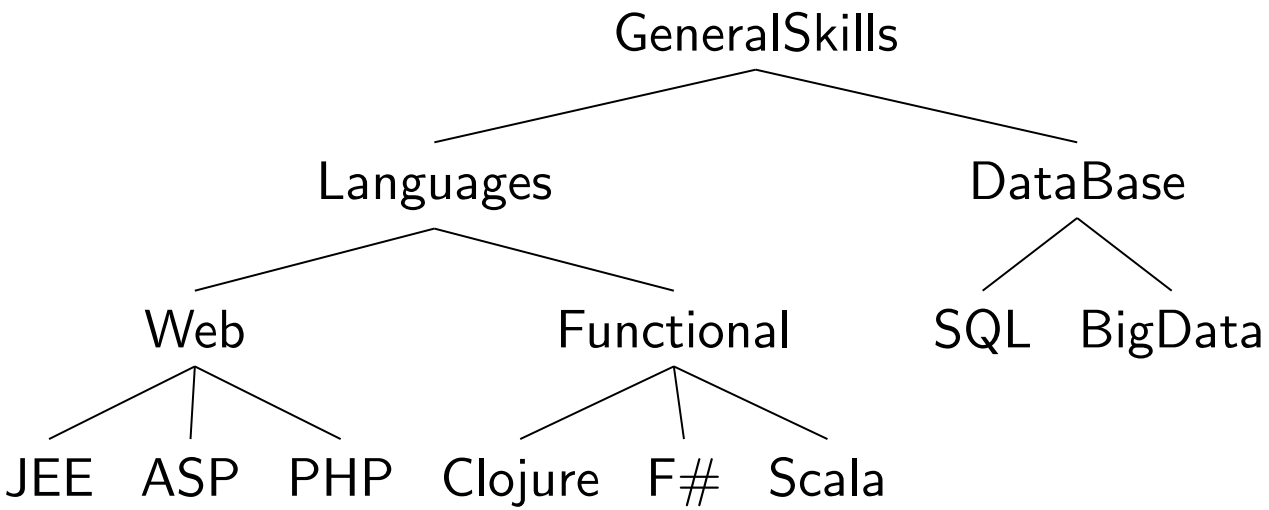
Profiles

$r = [ASP, BigData]$
$s_1 = [JEE, SQL]$
$s_2 = [Scala, BigData]$



Weightened Profiles

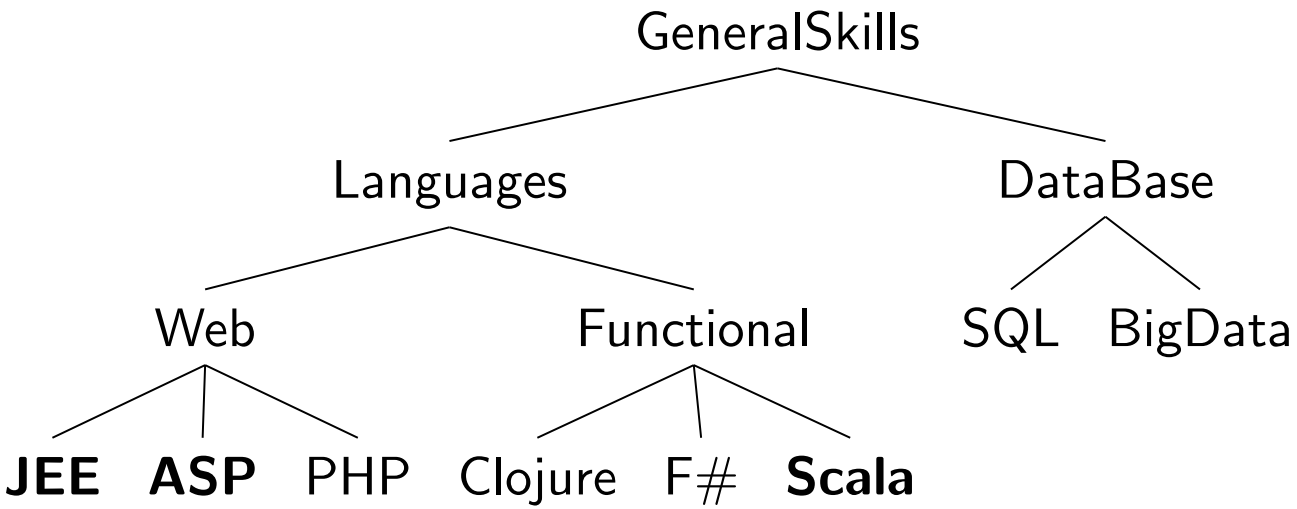
$r = [(ASP, 0.7),\ (BigData, 0.3)]$
$s_1 = [(JEE, 0.5),\ (SQL, 0.5)]$
$s_2 = [(Scala, 0.5),\ (BigData, 0.5)]$



Semantic Similarity

$$sim_c(c_1, c_2) = 1 - d_c(c_1, c_2)$$

$$sim_c(ASP, JEE) = 1 - d_c(ASP, JEE) = \frac{7}{8}$$
$$sim_c(ASP, Scala) = 1 - d_c(ASP, Scala) = \frac{5}{8}$$

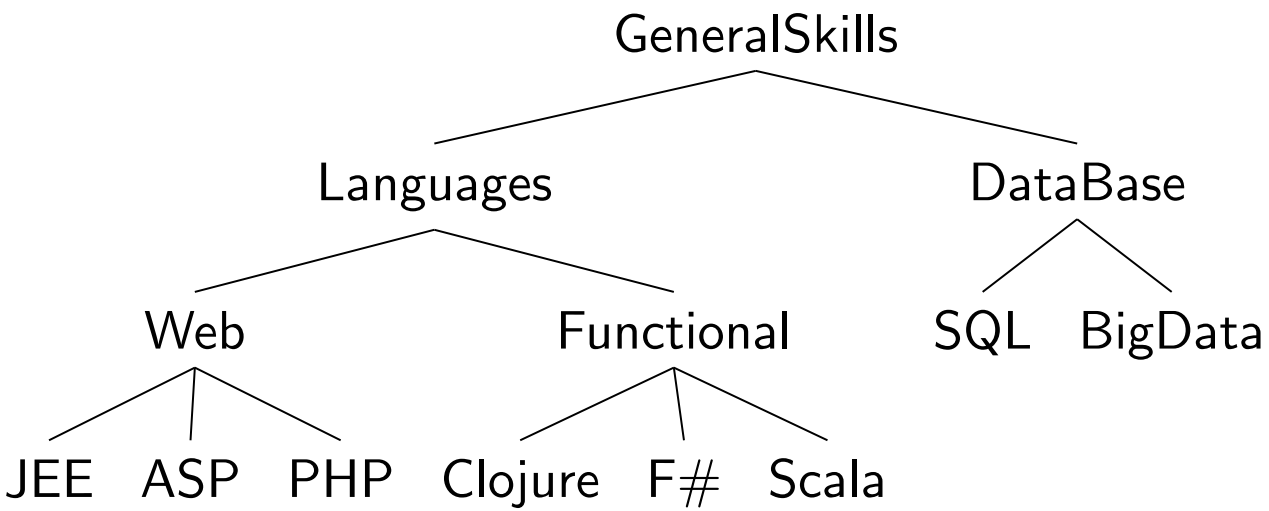


Semantic Distance

$$sim_c(c_1, c_2) = 1 - d_c(c_1, c_2)$$

$$d_c(c_1, c_2) = d_c(c_1, cpp) + d_c(c_2, cpp)$$
$$d_c(c_x, cpp) = milestone(cpp) - milestone(c_x)$$

$$milestone(c_x) = \frac{1}{2k^{level(c_x)}}$$



Competence Level Similarity

$$sim_p(cl_1, cl_2) = \begin{cases} 1 - \alpha(cl_1 - cl_2) & \text{if } cl_1 > cl_2 \\ 1 & \text{otherwise} \end{cases}$$

where $0 \leq \alpha \leq \frac{1}{4}$

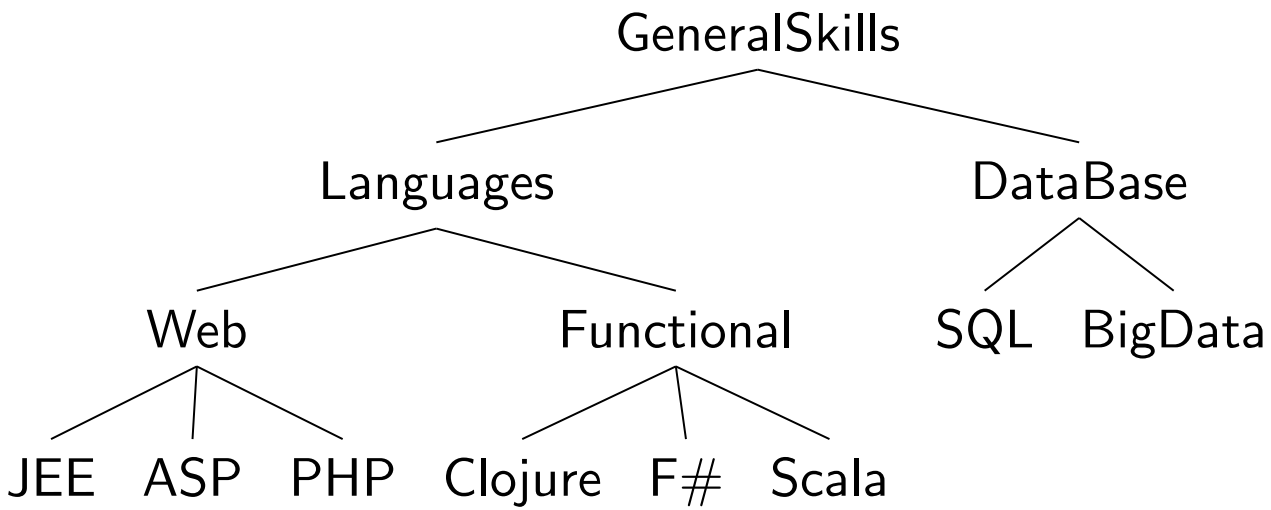
Global Similarity of Two Profiles

$$Sim(r,s) = \sum_i w(r_i).\max_j\{sim_c(r_i,s_j).sim_p(p_{r_i},p_{s_j})\}$$

$$\text{where } \sum w(r_i) = 1$$

Global Similarity of Two Profiles

$r = [(ASP, 0.7),\ (BigData, 0.3)]$
$s_1 = [(JEE, 0.5),\ (SQL, 0.5)] \quad \dashrightarrow Sim(r, s_1) \simeq 0.82$
$s_2 = [(Scala, 0.5),\ (BigData, 0.5)] \dashrightarrow Sim(r, s_2) \simeq 0.72$



Presentation Outline

Drawbacks of the Preceding Similarity

Asking for:

1. The weighting of the skills;
2. A competence level associated with each skill.

First Idea: With (2) inducing (1):

$$\textit{Sum of the weights } (\equiv 1) = \sum_i \textit{competenceLevel}(x_i)$$

$$\textit{weight}(x_j) = \frac{\textit{competenceLevel}(x_j)}{\sum_i \textit{competenceLevel}(x_i)}$$

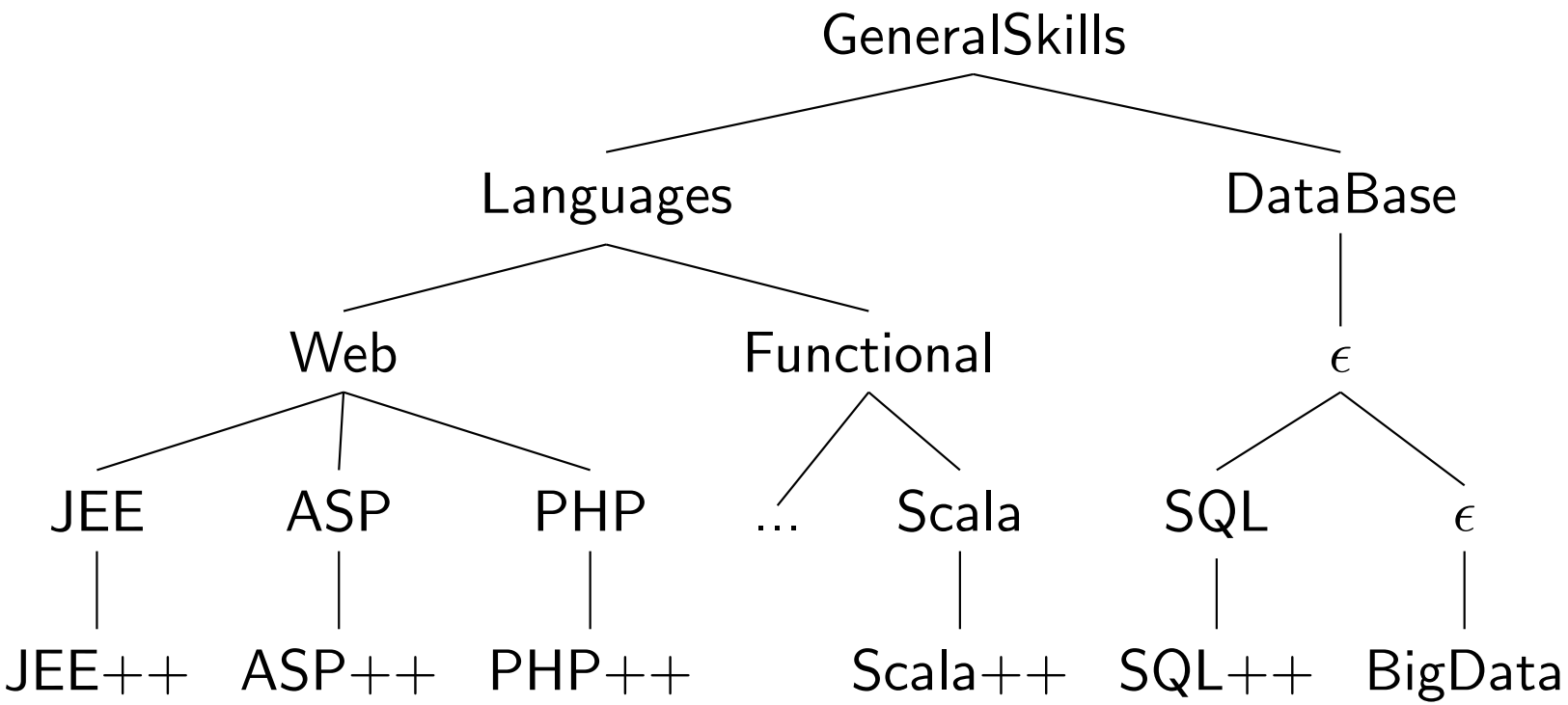
Levelized Taxonomy Approach

- Matching the inheritance relationship with the competence level;
- Matching the competence level with the depth of the hierarchy;
- $\Rightarrow$ Avoiding the weighting of the skills.

But,

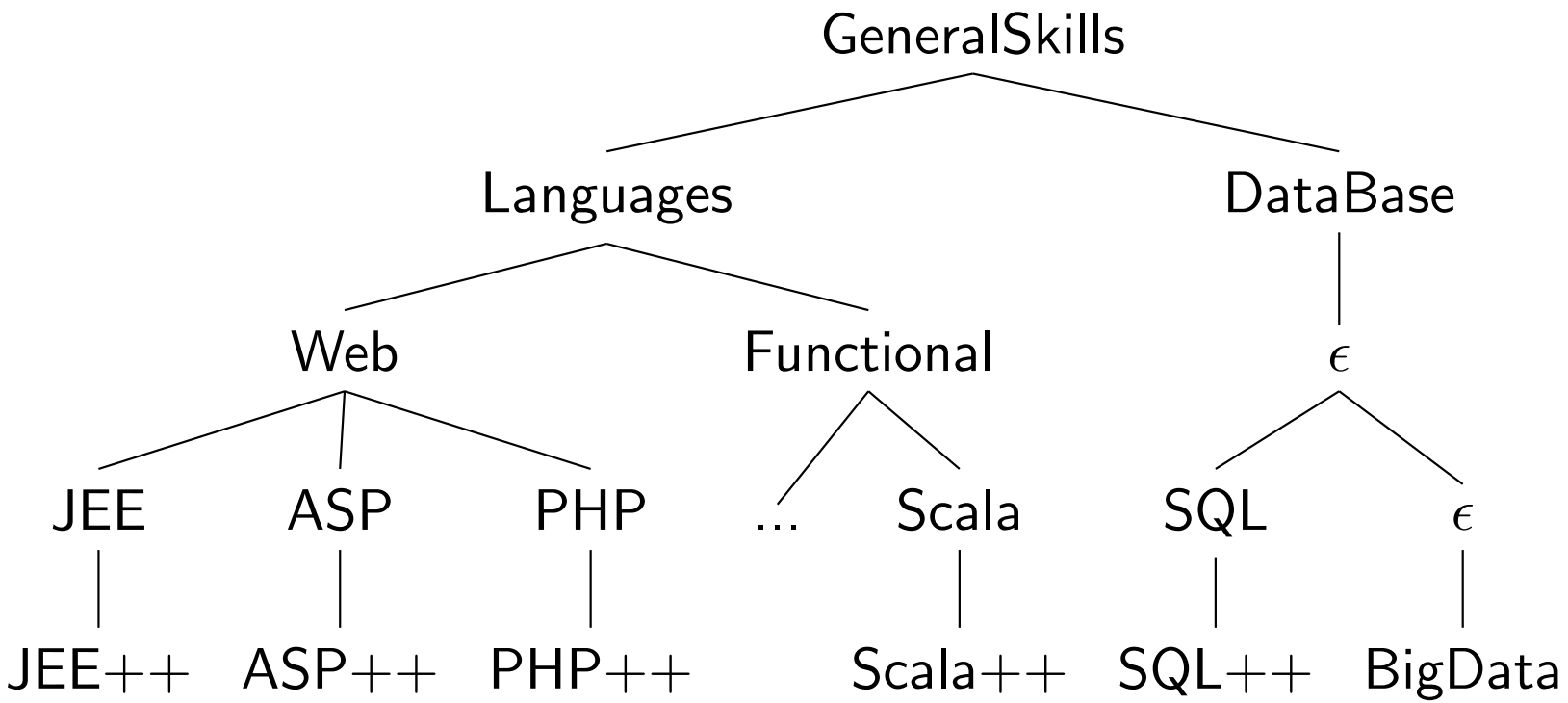
- More effort on the design of the taxonomy.
- Maybe: a loss of precision in the request.

Example of a Levelized Taxononomy



Global Similarity of Two Profiles

$r = [ASP++,\ BigData]$
$s_1 = [JEE++,\ SQL++]$ $\dashrightarrow Sim(r, s_1) \simeq 0.81$
$s_2 = [Scala++,\ BigData]$ $\dashrightarrow Sim(r, s_2) \simeq 0.78$



Levelized Taxonomy Approach

$$Sim(r,s) = \sum_i weight(r_i).\max_j\{sim_c(r_i,s_j).sim_p(level(r_i),level(s_j)\}$$

$$\text{where } weight(r_i) = \frac{level(r_i)}{\sum_k level(r_k)}$$

- Automating of the process.
- $\Rightarrow$ Less effort on the user side.
- More work on the conception side.

Presentation Outline

Conclusion

- Levelized Taxonomy: a new type of taxonomy which intends to bring more automaticity to the job recommender systems.

Future Work

- Methodology of conception.
- Practical tests on real datasets.
- Automatic building of levelized taxonomies.
- Extension of the similarity measure for more complex semantic networks.