Anton Andrejko

Mária Bieliková

Personalized Comparing Instances of Domain Ontology Concepts

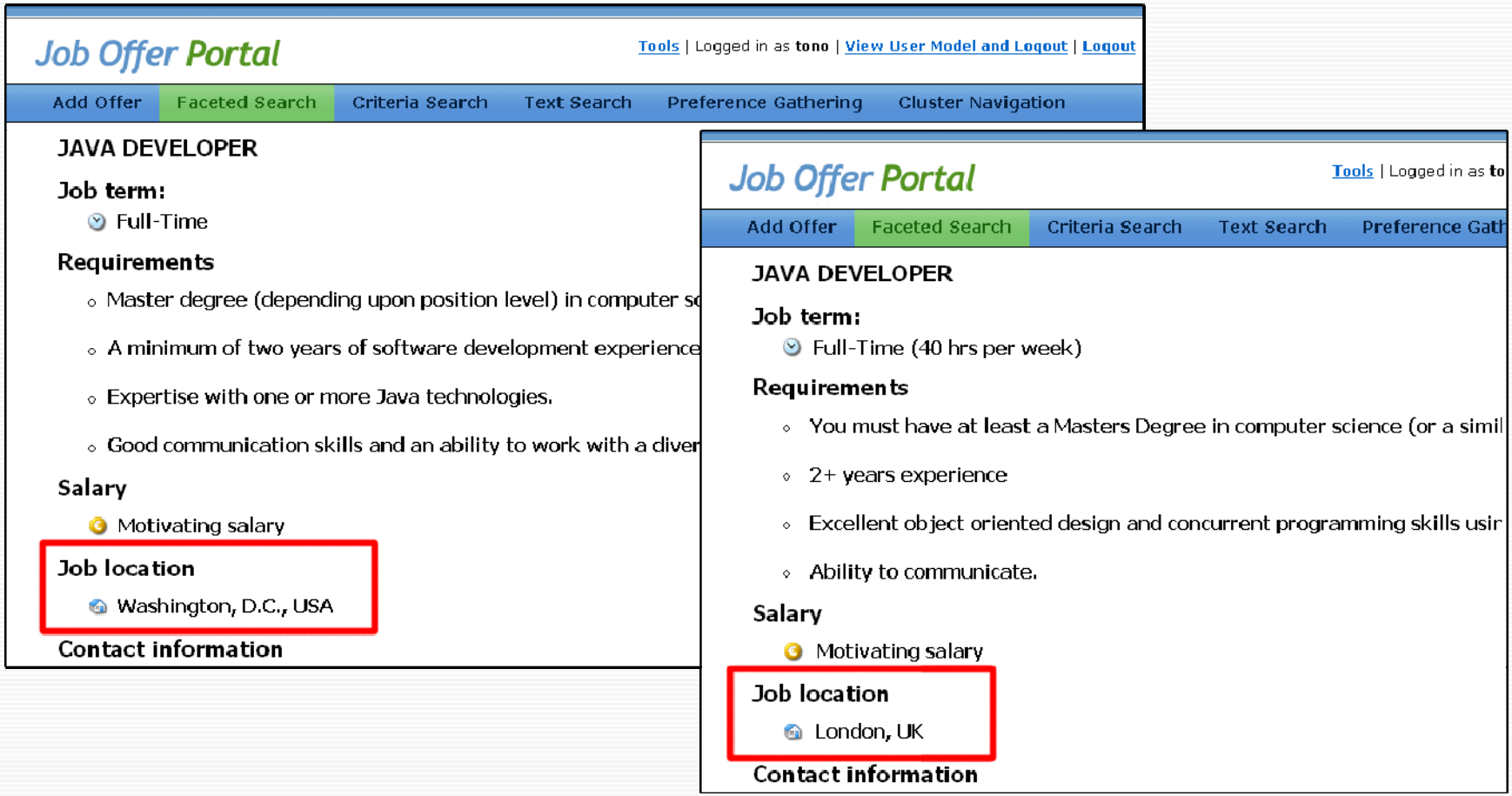
Institute of Informatics and Software Engineering

Faculty of Informatics and Information Technologies Slovak University of Technology

Motivation

- Personalization requires user model:
  - Populated with user characteristics
  - Reflecting the real user at the moment
- Sources of information:
  - The user
  - User's activity
  - Records on the web servers
  - **Content analysis**

Example on analysis of job offers

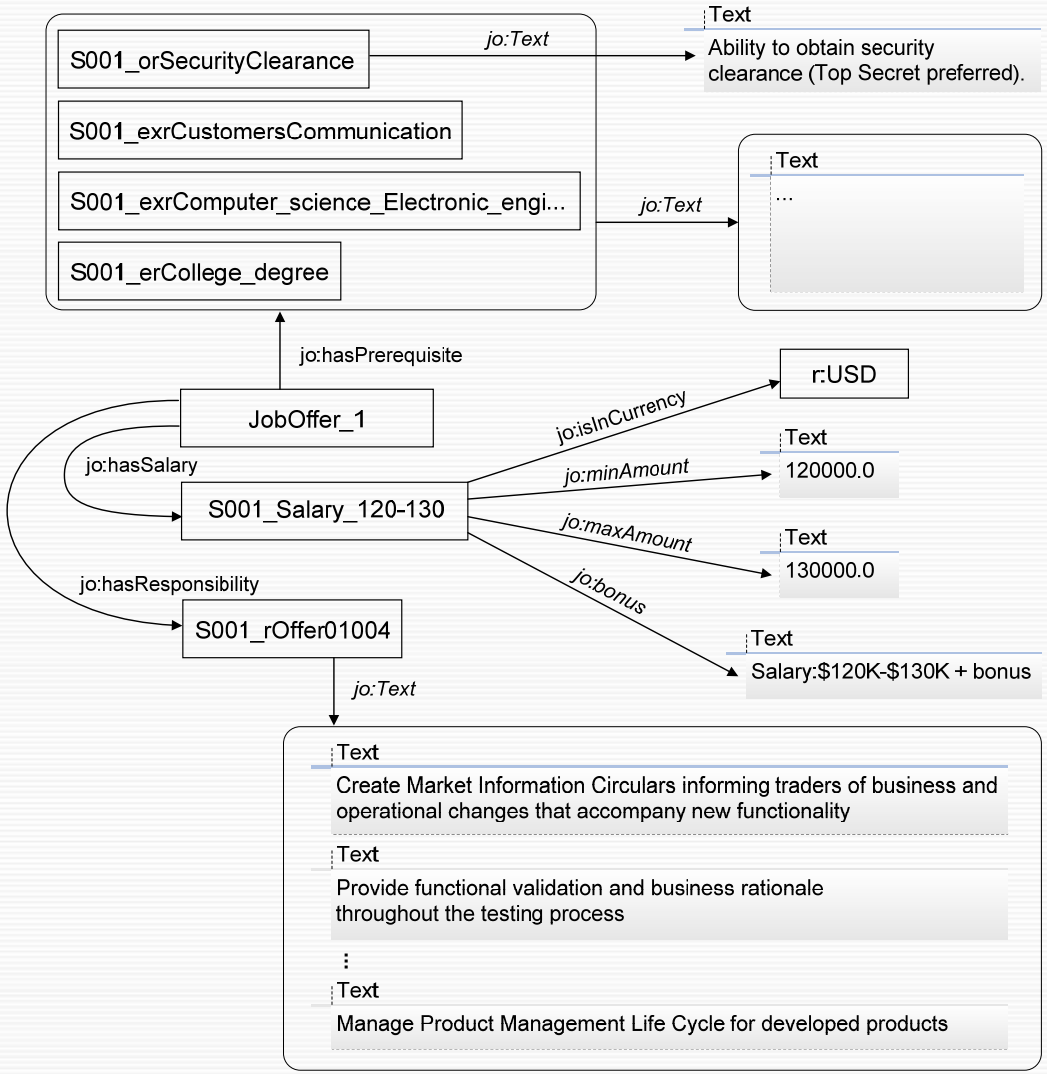


Our goals

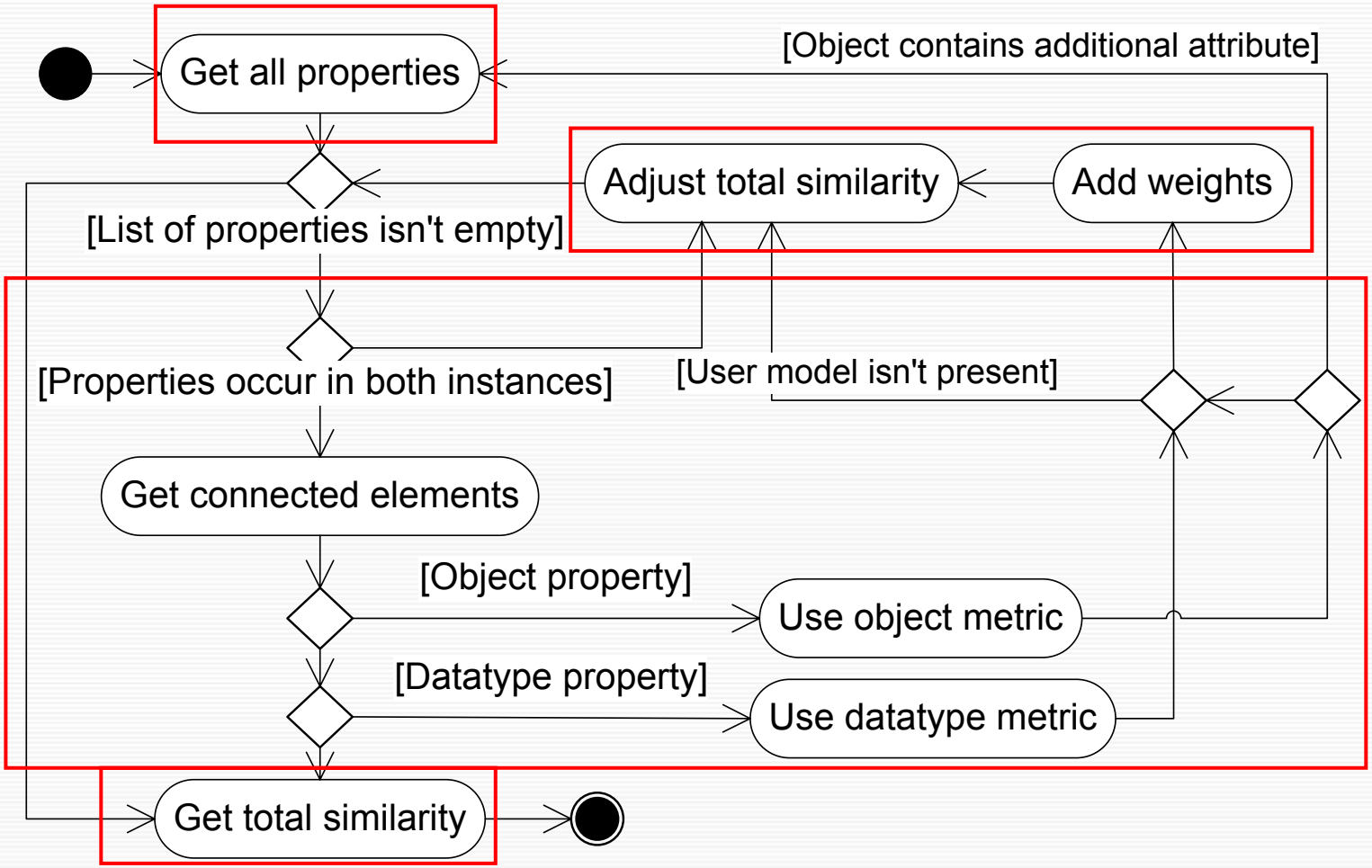
1. To compute similarity by using variety of similarity metrics
2. To acquire information for the user model from the computed similarity
3. To involve a user into computing similarity → personalized similarity

Example on job offer ontological instance

- Datatype and object properties
- Properties with different occurrences
- Different depth



Method principle



Acquisition of properties

- single occurrence in both instances
  - DATATYPE: datatype metrics with regard to the literals type is used
  - OBJECT: taxonomy distance and considering other properties➔ invoke recursive call of the method
  - Dealing with inverse properties (e.g., *isDutyLocationOf*) and loops
- single/multiple occurrence in one instance only
  - DATATYPE & OBJECT: adjusting total similarity
- multiple occurrence in both instances
  - DATATYPE & OBJECT: identification of “matching” pairs

Similarity estimation

$$sim(InstA, InstB) = \frac{\sum_{i=0}^{|A \cap B|} \text{PropertySM}_i(elementA, elementB)}{|A \cup B|}$$

- The total similarity is a mean aggregate of partial similarities where:
  - *InstA, InstB* – compared instances of ontological concepts
  - A, B – sets of properties which compared instances consist of
  - *PropertySM* – similarity measure that computes the similarity between elements according to a metric
  - *elementA and elementB are the elements (instances or literals) connected to the particular property*
- **The aggregate of partial similarities is always the same no matter what the context is**

Personalized extension to similarity

$$\mathrm{sim}(InstA, InstB) = \frac{\sum_{i=0}^{|A \cap B|} weight_i \times \mathrm{PropertySM}_i(elementA, elementB)}{\sum weight}$$

- We introduce weights to compute similarity for individual users
- Proposed weights:
  - *"1" if there is no correlation between a property of the instance and a characteristic in the user model,*
  - *"w" if there is match not only between a property of the instance and a characteristic but also between their values,*
  - *a value between the previous two values means that there is a match between the examined property of the instance and the user model, but the related value is not identical.*

Characteristics vs. preferences

- Each **characteristic** is assigned a **value** when initialized
- **Preference** indicates that the related property is important for the user but there is no specific value assigned to it
  - *weight = w/2*
- Characteristics and preferences
  - *weight = w + w/2*

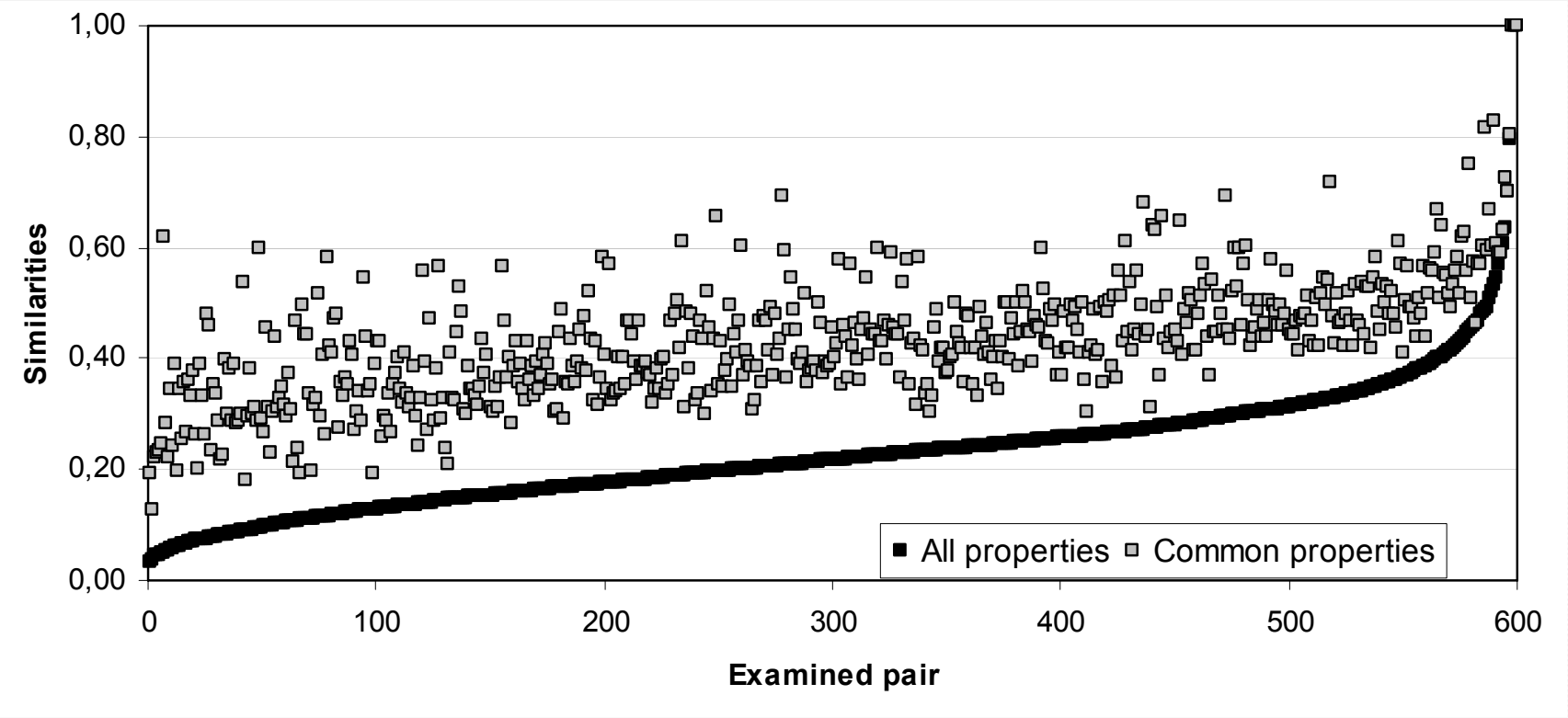
Content analysis

- We assume that a property that the user likes will likely influence his or her rating towards higher (or positive) values and vice versa.
- Explicitly expressed user's interest → characteristics for the user model
- 2 thresholds (positive and negative) were set experimentally using Pareto principle on 55 000 properties
  - Positive set → if similarity computed for a property (e.g., hasJobTerm) > 0,65
  - Negative set → if similarity computed for a property < 0,25
- Sets of properties are transformed into user characteristics by Log Analyzer tool

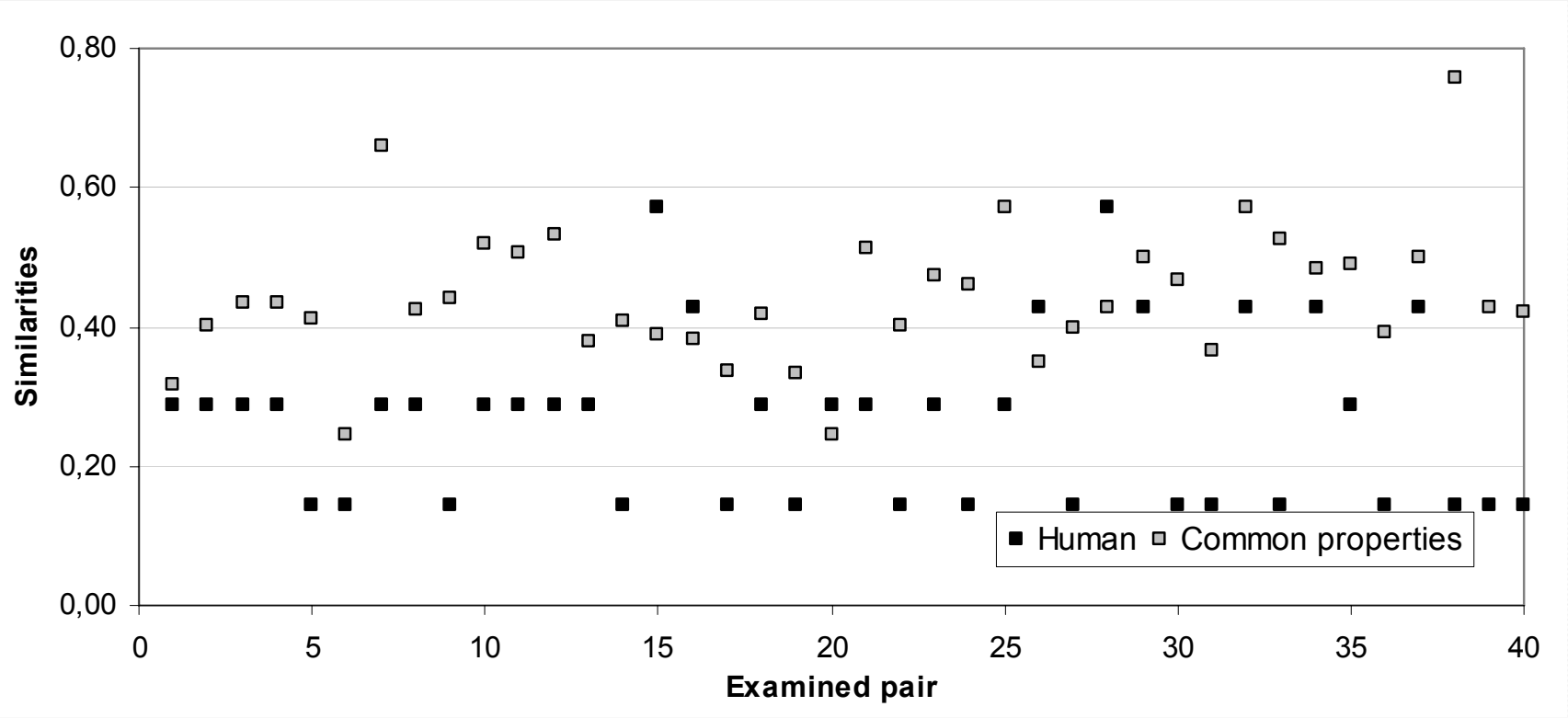
Experimental evaluation

- Job offers application domain
  - developed in the course of the research project NAZOU (Tools for acquisition, organization and maintenance of knowledge in an environment of heterogeneous information resources, http://nazou.fiit.stuba.sk)
  - similarity computed for 10 000 pairs
- Software prototype ConCom computes similarity for:
  - all properties or
  - common properties only
- Implementation
  - Java
  - Sesame repository

Experimental evaluation



Experimental evaluation



Conclusions

- Introducing (personalized) similarity computed with regard to the user model
- Investigating causes of similarity according to positive and negative thresholds
- Similarity is computed for particular properties
  - considering all properties is better to investigate properties that influenced user's evaluation
  - the similarity where only common properties are considered is more suitable if the user model is involved
- Using method to recommend similar content

Future work

- Case study with the users to quantitatively evaluate personalized similarity
- Experimenting with other application domains (e.g., scientific publications)
- Using methods in other fields (e.g., clustering, recommender systems)