jobsta

Study - Job - Money

Project for
Introduction to Semantic Systems
(188.399-2019W)

Group 01

Cem Bicer (01425692)
Helmuth Breitenfellner (08725866)
Laszlo Kiraly (09227679)
Gerald Weber (00125536)

2020-01-31

App Idea

jobsta

Study - Job - Money

- For Software Developers and Data Scientists
- Asks for experience, age, location
- Answers to following questions:
  - *What shall I study?*
  - *Where shall I work?*
  - *What shall I practise?*
  - *How can I improve?*

The Mobile App

Data Sources

- **Kaggle User Survey**
  Data Scientists, Country, Job Role, Programming Language, Income
- **StackOverflow User Survey**
  Software Developer, Country, Job Role, Programming Language, Income
- **GitHub Repositories Data**
  Repository URL, Popularity, Programming Language, Issues
- **TISS Lectures**
  Lectures, Lecturer, Description, Programming Language

Kaggle Survey

- https://www.kaggle.com/c/kaggle-survey-2019
- Used Jupyter Notebook for Pre-Processing
- Created RDF directly from Python
- **Challenge:** high number of one-hot-encoded values, had to extract unique values

StackOverflow Survey

- https://insights.stackoverflow.com/survey/2018
- Used Python for Pre-Processing
- Created RDF directly from Python
- **Challenge:** Excel, Numbers and TextMate could all not open the csv file (>90.000 entries) properly

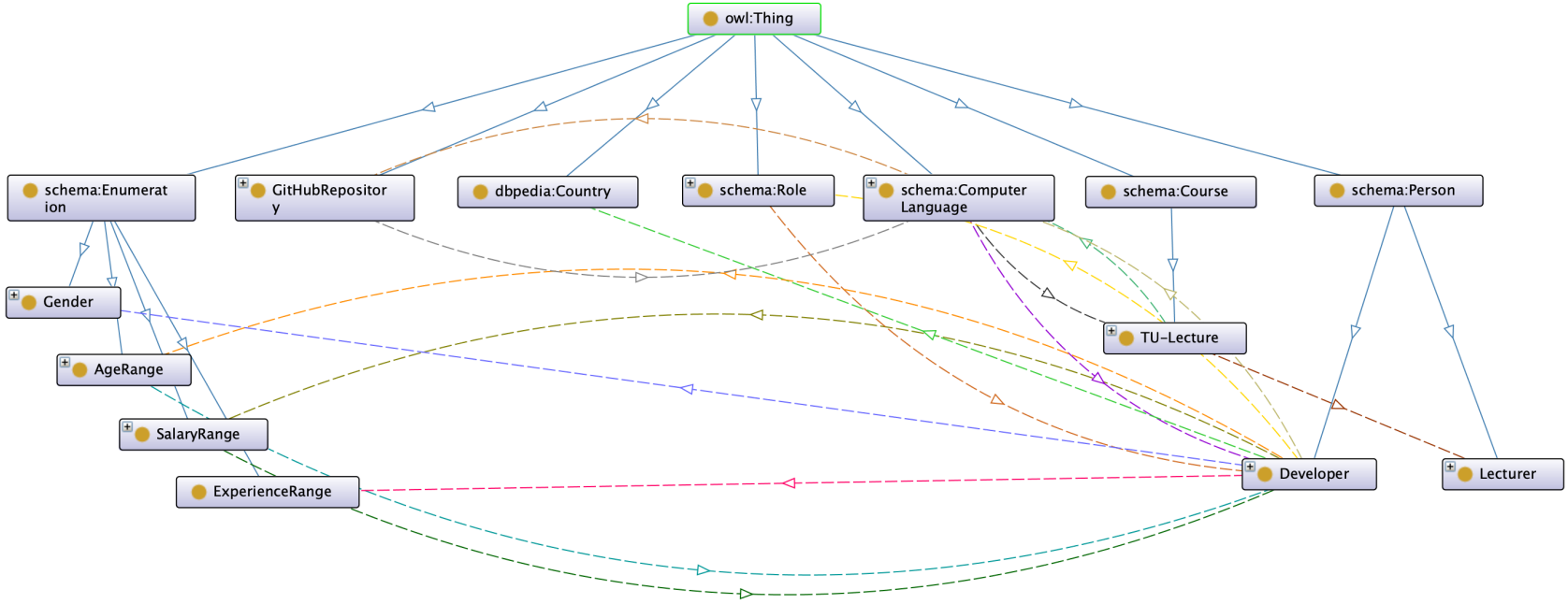
GitHub Repositories Data

- http://ghtorrent.org/
- Used Bash & R Script for Pre-Processing
- Created RDF directly from Python
- **Challenge:** *huge* data archive (>100GB) had to be filtered / preprocessed

TISS Lectures

- https://tiss.tuwien.ac.at/course/courseList.xhtml?dswid=6403&dsrid=238
- Used Python Script
- Created RDF directly from Python (using `rdflib`)
- **Challenge:** web scraping, identifying the programming language from text

Ontology #1


owl:Thing
schema:Enumeration
GitHubRepository
dbpedia:Country
schema:Role
schema:ComputerLanguage
schema:Course
schema:Person
Gender
AgeRange
SalaryRange
ExperienceRange
TU-Lecture
Developer
Lecturer

Ontology #2

- Created with Protégé
- Reusing existing Ontologies
  - schema.org
  - dbpedia.org
- Entites:
  - Developer
  - dbpedia:Country
  - schema:Course
  - GitHubRepository
  - schema:ComputerLanguage
  - ...
- Object properties:
  - dealsWith
  - developsIn
  - schema:homeLocation
  - ...

Harmonize Data I

- Age Ranges
  - Different Age Ranges
- Salary vs. Salary Range
  - Salary Range in Kaggle
  - Salary Value in Stackoverflow
- Roles
  - Combined from Surveys into List
  - e.g. Frontend Developer -> Software Engineer
  - ... C-Suite Executive -> Manager

Harmonize Data II

- Countries
  - dbpedia linked to external data
- Gender
  - Single Selection in Kaggle
  - Multiple Selections in Stackoverflow
- Computer Language
  - Combined from Surveys into List
  - Field in Github Repository
  - Extracted from TISS Lecture Description

SPARQL Queries #1

"As a developer I live in (Austria) and I want more than (150000 USD per year). What courses at TU Wien deal with programming languages which high-earners are using?"

SPARQL Queries #1 (cont.)

```
PREFIX group1: <http://www.semanticweb.org/sws/ws2019/group1#>
PREFIX schema: <http://schema.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dbpedia: <http://dbpedia.org/resource/>

SELECT DISTINCT ?lecture ?language
WHERE {
  ?lecture group1:dealsWith ?language .
  ?developer group1:developsIn ?language .
  ?salaryRange group1:maxSalary ?salary .
  ?developer group1:hasSalaryRange ?salaryRange .
  ?developer schema:homeLocation dbpedia:Austria .
  FILTER (?salary > "150000"^^xsd:integer)
}
LIMIT 25
```

SPARQL Queries #2

"As a developer I live in (Austria) and I can program in (Python) and I want more than (55000 USD per year). Should I stay or should I go?"

SPARQL Queries #2 (cont.)

```
ASK
WHERE {
  ?developer a group1:Developer .
  ?developer schema:homeLocation ?country .
  ?developer group1:developsIn ?language .
  {
    SELECT ?country (AVG(?avgRange) as ?averageK)
    WHERE {
        ?developer a group1:Developer .
        ?developer schema:homeLocation ?country .
        ?developer group1:developsIn ?language .
        ?developer group1:hasSalaryRange ?salaryRange .
        ?salaryRange group1:minSalary ?minSalary .
        ?salaryRange group1:maxSalary ?maxSalary.
        BIND ((?minSalary + ?maxSalary)/2 AS ?avgRange)
    }
    GROUP BY ?country
  }
  {
    SELECT ?country (AVG(?salaryValue) as ?averageS)
    WHERE {
      ?developer a group1:Developer .
      ?developer schema:homeLocation ?country .
      ?developer group1:developsIn ?language .
      ?developer group1:salary ?salaryValue .
    }
    GROUP BY ?country
  }
  BIND ((?averageK + ?averageS)/2 as ?average)
  FILTER (?language = group1:Python && ?country = dbpedia:Austria && ?average > "55000"^^xsd:integer)
}
GROUP BY ?country ?average
```

Lessons Learned

- Iterative process to come up with final idea
- Scraping TISS: no ID access to fields
- http://schema.org not equal to http://www.schema.org
- GraphQL Github API vs. Database Dump
- Harmonizing data can be tedious

Questions?

Thank you for your attention!