(Meta-)Datamanagement with KNIME

SWIB 2017 Workshop
Your mentors

Prof. Dr. Kai Eckert

- Stuttgart Media University
- Focus: web-based informations systems

Prof. Magnus Pfeffer

- Stuttgart Media University
- Focus: information management
Current projects with data focus

Specialised information service for Jewish studies

Challenges:

- Integration of heterogeneous datasets
- Contextualization using external sources
- Merging data across language and script barriers
Current projects with data focus

Linked Open Citation Database

Challenges:

- Bad data
  - ... OCRRed references...
  - ... created by the authors...
- Identity resolution
- Complex data model
- Natural Language Processing

Consortium

Funding by

DFG

UB MANNHEIM

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Hochschule der Medien

ZBW

Leibniz-Informationszentrum
Wirtschaft
Leibniz Information Centre for Economics
Current projects with data focus

Japanese visual media graph (funding pending…)

Challenges:

- Multitude of entities and relations
  - Work, release, adaption, continuation
  - Creators, producers, staff, actors
  - Characters
- No traditional data sources (libraries, etc.)
- Fan-produced data is the best available source

Consortium
Today’s Workshop

- Part 1: Introduction (~ 2 hrs)
  - Installation and preparation
  - Basic concepts
  - Basic data workflow
    - Loading
    - Filtering
    - Aggregation
    - Analysis and visualization
  - Advanced workflow
    - Dealing with errors and missing values
    - Enriching data
    - Using maps for visualization
Today’s Workshop

● Part 2: Real-world uses (~ 1 hr)
  ○ Using the RDF nodes to read and output linked data
  ○ Creating an enriched bibliographic dataset
    ■ Fixing errors in the input dataset
    ■ Downloading bibliographic data as XML from the web
    ■ Enriching with classification data from a different source
    ■ Data output

● Part 3: Data challenge
  ○ Did you bring interesting data? Do you have any specific needs?
Part 1: Introduction
Installation

- Please chose the 64bit version whenever possible
- KNIME:// protocol support must be activated
- Use the full package, so there is no need to download modules later
Installation

- Watch out for the memory settings, allot enough memory to KNIME
- Can be changed by editing the config file KNIME.ini
Why KNIME?

Possible alternative: Develop own software tools?

Upside: Maximum flexibility

Downsides:

● Very complex, coding knowledge a necessity
● Own code can get messy, hard to maintain and document
● Shared development can lead to friction and overhead
● Modules and standard libraries often do not cover all aspects

→ Maybe it is better to use an existing toolset for metadata management
Why KNIME?

Alternative: Toolsets?

Some exist:

- Simple command-line tools and tool collections
- Catmandu
- Metafacture

→ Single tools are very inflexible

→ Toolsets are still quite complex, need coding proficiency and still are very challenging for new users

→ So maybe an application-type software would be better?
Why KNIME?

Alternative: Application software for data management?

Examples:

- OpenRefine
- d:swarm

→ Easy access, but limited functionality
→ Fixed workflow (OpenRefine) or fixed management domain (d:swarm)
→ Extensions are hard to do
That is why KNIME

Open source version available (extra functionality requires licensing)

GUI-driven data management application

Supports multiple types of different workflows

Very good documentation, self-learning support for newcomers

Many extensions exist, and creating your own is well supported

Development in a team or using other people’s data workflows is integral to the software
Workflows

Classic data workflow: Extract, Transform, Load (ETL)

KNIME adds:

Extensions for analysis and visualization

Extensions for machine learning

...and much more
KNIME GUI

workspace management

active workspace

documentation

node selection

logs
Nodes

Basic KNIME idea: nodes in a graph form a “data pipeline”

- Nodes for all kinds of functions
- Configuration is done using the GUI
- Directed links connect nodes to each other
- Processing follows the links
- Transparent processing status
  - Red: inactive and not configured
  - Yellow: configured, but not executed
  - Green: executed successfully
Example: “Data Blending”

Local example workflow included in the KNIME distribution

KNIME://LOCAL/Example%20Workflows/Basic%20Examples/Data%20Blending

(Demo)
Example: a simple ETL workflow

Login to the EXAMPLES server of KNIME

right mouse button
Example: ETL Basics

KNIME://EXAMPLES/02_ETL_Data_Manipulation/00_Basic_Examples/02_ETL_Basics

(Demo)
My first workflows

Generate some data (Excel or LibreOffice)

● Columns author, title, year, publisher
● 3-4 sample datasets
● Save as both CSV file and Excel spreadsheet

In KNIME:

● Use a file node to open the CSV file
● Use a filter node to limit columns to title and year
● Use a filter node to select only those rows where year > 2000
● Use a file node to save the result as a CSV file
My first workflows

We prepared an XML file with data on the TOP 250 entries of IMDB.com (movies.xml)

In KNIME:

- Preparation: Open the file, create a table from XML data
- Filter 1: Only title and year information
- Filter 2: All information on films from 2012
- Filter 3: What are the titles of the films from the years 2000-2010?
- Analysis 1: What genres are contained in the file?
- Analysis 2: Which director appears most often?
Example: Data visualization

Example data visualization.knwf

(Demo)

knime://EXAMPLES/03_Visualization/02_JavaScript/04_Example_for_JS_Bar_Chart

(Demo)
My first visualization

Using movies.xml

In KNIME:

- Determine the countries, in which the movies take place and count their occurrence
- Use a pie chart to show the numbers
- Use a bar chart to show the numbers

Advanced exercise: What information is missing to visualize the countries as discs on a world map, with the size of the disc corresponding to the number?
Using external sources to enrich data

json demo.knwf

(Demo)
Using external sources to enrich data

Using web APIs

KNIME://EXAMPLES/01_Data_Access/05_REST_Web_Services/01_Data_API_Using_REST_Nodes

(Demo)
My first enrichment

Have address, want geo-coordinates? Geocoding!

https://developers.google.com/maps/documentation/geocoding/start

In KNIME:

- Extend the list of countries to contain an URL for the google API
- Use the GET-node and query google
  - Warning: there is a rate control on the google APIs!
  - Use the node configuration to slow down the queries

Did we get correct coordinates for all countries? How did you check?
Example geo-visualization

KNIME://EXAMPLES/03_Visualization/04_Geolocation/04_Visualization_of_the_World_Cities_using_Open_Street_Map_(OSM)

(Demo)
Using geo-visualization

Again using movies.xml

In KNIME:

- visualize the countries that the movies are taking place in as discs on a world map, with the size of the disc corresponding to the number
Part 2: RDF and a real-world example
RDF in KNIME
Node group: Semantic Web/Linked Data

- Memory Endpoint as internal storage
- SPARQL Endpoint to read/write data

- IO is very basic:
  - Triples from tables to/from file
  - Triples from graphs to/from file

- Important table structure: subj, pred, obj

- Free SPARQL queries can be used to query for additional data.
- RDF data manipulation
Consuming RDF in KNIME

knime://EXAMPLES/08_Other_Analytics_Types/06_Semantic_Web/11_Semantic_Web_Analysis_Accessing_DBpedia

(DEMO)
Use the right tools!

knime://EXAMPLES/08_Other_Analytics_TYPES/06_Semantic_Web/10_Using_Semantic_Web_to_generate_Simpsons_TagCloud

(DEMO)

Fixed version:
10_Using_Semantic_Web_to_generate_Simpsons_TagCloud_FIXED.knwf

- The demo needs some fixes to actually get the word cloud.

- Most part of the workflow is about trimming and filtering RDF strings (e.g., get rid of the xsd types).

- It is great that it is possible to do this in KNIME, but the creation of a proper CSV file outside KNIME might be easier.
Producing RDF in KNIME

Use your movie workflow to produce triples for title and year of a movie.

Approach:

1. Create a column `subj` containing the subject of each row
2. For each predicate to be written:
   a. rename the column containing the value to `obj`
   b. add a column `pred` containing the desired property.
   c. filter to keep only the columns `sub`, `pred`, `obj`
3. Concatenate the resulting tables (or write them to a triple store)
Producing RDF in KNIME

( DEMO ) Kurs_Movies_Filter_With_RDF.knwf

Again the question: Is creating triples from CSV outside KNIME easier?
Case Study: Metadata enrichment
Input

- A table of library holdings:
  - Item number and barcode to identify an item.
  - PPN to identify the manifestation of each item.
  - Call number (Signatur) and location (Sigel) for each item.

- No metadata!

- Goal: Get classification data (RVK) for each item.

![Table](image)
Output

1. Group per PPN
2. Add Metadata from SWB union catalog.
3. For entries without RVK: Add RVKs from BVB.
4. Modify result table to match required CSV format.
   (This workflow ends here!)
5. Data is then processed in another application to do manual quality checks and add additional RVK.
6. Afterwards, there is another workflow to ungroup back to item level.
Group/Ungroup

- A typical step is to switch the levels of aggregation to make use of KNIME operators.
- Here is an example where a row filter is used to actually filter elements of a list element (“Remove Non-RVK” in the workflow):
Looping over rows

- When the workflow was created, the GET operator could retrieve data for a whole table, but if one request failed, the whole operator failed and the workflow stopped.
- Moreover, the GET operator did not pass through other columns than the URL columns.
- Both problems are dealt with in the SWB fetch part:
  - A loop is created over all rows.
  - The resulting table with (additional) columns is joined with the original table.
Deal with empty results

- Sometimes whole parts of the workflow can be skipped.

- Example: We filter for all rows who have no RVK but have author and title information available (as we need this to search for matching records).

- Depending on the (sampled) input data, there might be no rows who qualify. Then we just bypass the whole RVK enrichment part of the workflow.