Programming Technologies for Web Resource Mining

SoftLang Team, University of Koblenz-Landau

Prof. Dr. Ralf Lämmel
Msc. Johannes Härtel
Msc. Marcel Heinz

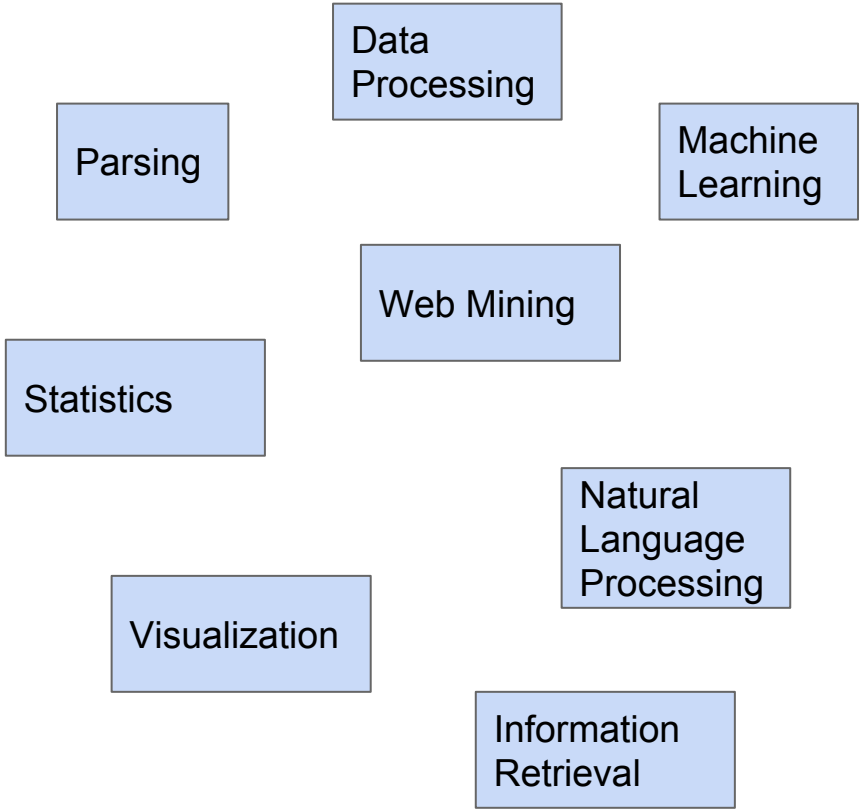
Motivation

- What are interesting web resources?
  - ?
- What can web resources be used for?
  - ?
- In what form do we find data?
  - ?

Motivation

- What are interesting web resources?
  - Online Code Repositories (Github, Bitbucket)
  - Wikipedia
  - Customer Data
  - Facebook/Twitter, etc.
- What can web resources be used for?
  - Recommender Systems
  - Social Analysis
  - Knowledge Engineering
- In what form do we find data?
  - Computed by Web Services (REST)
  - Queried through SPARQL endpoints.
  - Downloadable as dumps
    - Huge data sizes
    - Compressed data formats
    - Download with Torrents (?)

Disciplinary Overview



Data Processing

- Data Sources

Aggarwal, Charu C. *Data mining: the textbook*. Springer, 2015

- Documents indexed on the web.
  - Texts
  - Semi-structured Overviews.
- Access logs.
- Customer behavior profiles.
- Link structure on the web.

Data Processing

- Data Preparation

Aggarwal, Charu C. *Data mining: the textbook*. Springer, 2015

- **Raw data** may exist in a format that is unsuitable for automatic processing.
- A raw format has to be **translated** into a format that is suitable.

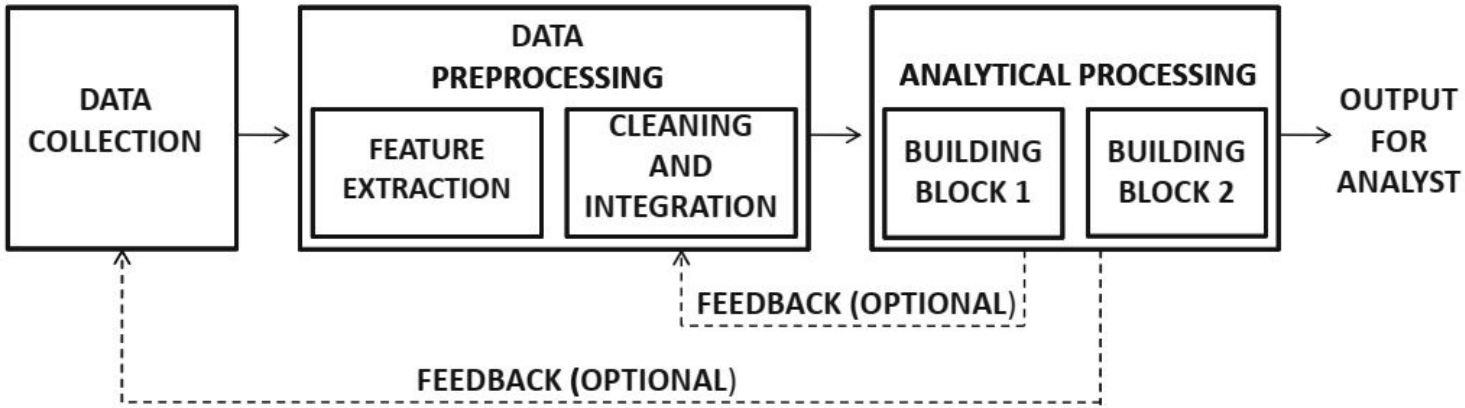


Figure 1.1: The data processing pipeline

Data Processing

Aggarwal, Charu C. *Data mining: the textbook*. Springer, 2015

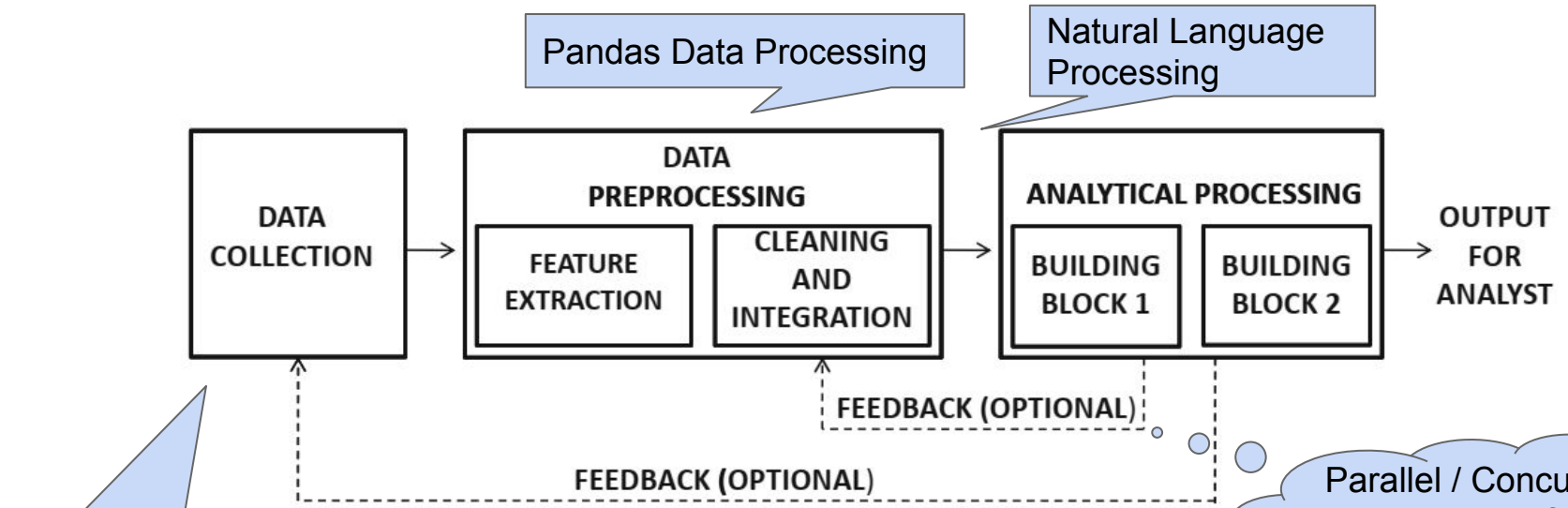


Figure 1.1: The data processing pipeline

Knowledge on the Web

- Wikipedia
  - is the biggest encyclopedia online.
  - is collaboratively edited and maintained.
  - is maintained by non-experts and experts.
  - has several complex mechanisms for quality assurance, e.g. through bots.
- DBpedia
  - is a structured mirror of Wikipedia.
  - mainly contains structured information in the triple format.
  - contains triples extracted from Wikipedia.
- Wikidata
  - is a general ontology.
  - structures knowledge into pure triples from every field of study.
  - is collaboratively edited and maintained.

Wikipedia

https://en.wikipedia.org/wiki/Java_(programming_language)

Java (programming language)

Title contains disambiguation containing "language"

Programming Language Infobox

Java is a general-purpose [computer-programming language]

Hypernym language in first sentence

The aim of this **list of programming languages** is to include all notable [programming languages] in existence

- Java
- JavaFX Script
- JavaScript
- JCL
- JEAN
- Join Java
- JOSS
- Joule

Included in an article on a list of software languages

Paradigm	Multi-paradigm: object-oriented (class-based), structured, imperative, generic, reflective, concurrent
Designed by	James Gosling
	n Microsystems (now ned by Oracle Corporation)
	ay 23, 1995; 23 years ago[1]
Typing discipline	Static, strong, safe, nominative, manifest
License	GNU General Public License, Java Community Process

DBpedia

http://live.dbpedia.org/page/Java_(programming_language)

- Wikipedia's data is preprocessed and translated into RDF data.
  - This means everything is saved as subject property object.
  - Read more on RDF at https://www.w3.org/RDF/
- Every page is treated as a resource with its link being its unique identifier. (e.g., **http://dbpedia.org/resource/Java_(programming_language)**)
- A Wikipedia article's summary is associated through the property dbo:abstract.
- The original Wikipedia article link from which the data was extracted is associated through the property dbo:wikiPageRevisionLink.
- The infobox template being used is associated through dbp:wikiPageUsesTemplate.

Wikidata

https://www.wikidata.org/wiki/Q251

- Knowledge is encoded in RDF data.
  - This means everything is saved as subject property object.
  - RDF is used to share knowledge in a machine readable format.
  - Read more on RDF at https://www.w3.org/RDF/.
- Wikidata is an ontology.
  - The data is validated against the schema, where constraints can be expressed.
  - Subjects, Properties and Objects are typed.
- Pure text blocks are avoided.

Assignment 3

Wikipedia Access – Web API

see https://www.mediawiki.org/wiki/API:Main_page

- Documents can be accessed through Endpoints.
- HTTP is used for communicating.
- Wikipedia's Endpoint/API:
  - https://en.wikipedia.org/w/api.php
- For Java (programming language):
  - Metainfo: https://en.wikipedia.org/w/api.php?action=query&titles=Java_(programming_language)
  - Page content: https://en.wikipedia.org/w/api.php?action=query&titles=Java_(programming_language)&prop=revisions&rvprop=content

REST

- REpresentational State Transfer

- API - Application Programming Interface.
  - An interface for functionality that is reusable by programs.

- Web API.
  - An interface provided over the web (using HTTP).

REST

Wikipedia implements REST

Aspects of a RESTful web service API

- A base URI such as http://example.com/resources/.
- An internet media type such as JSON.
- Standard HTTP methods: GET, PUT, DELETE, POST
- Links to reference state such as http://.../resources/1.

REST

https://en.wikipedia.org/wiki/Representational_state_transfer

Constraints of REST - Representational State Transfer

- “Client-Server”: Data on server, UI on client
- “Stateless”: No **client context** is stored on the server.
  - The data on the server can change and may thus not be viewed as stateless.
- “Cacheable”: Responses define themselves as cacheable.

REST is an architectural style - not a protocol!

REST – HTTP Messages

See https://en.wikipedia.org/wiki/Representational_state_transfer

Uniform Resource Locator (URL)	GET	PUT	PATCH	POST	DELETE
Collection, such as `https://api.example.com/resources/`	**List** the URIs and perhaps other details of the collection's members.	**Replace** the entire collection with another collection.	Not generally used	**Create** a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.[17]	**Delete** the entire collection.

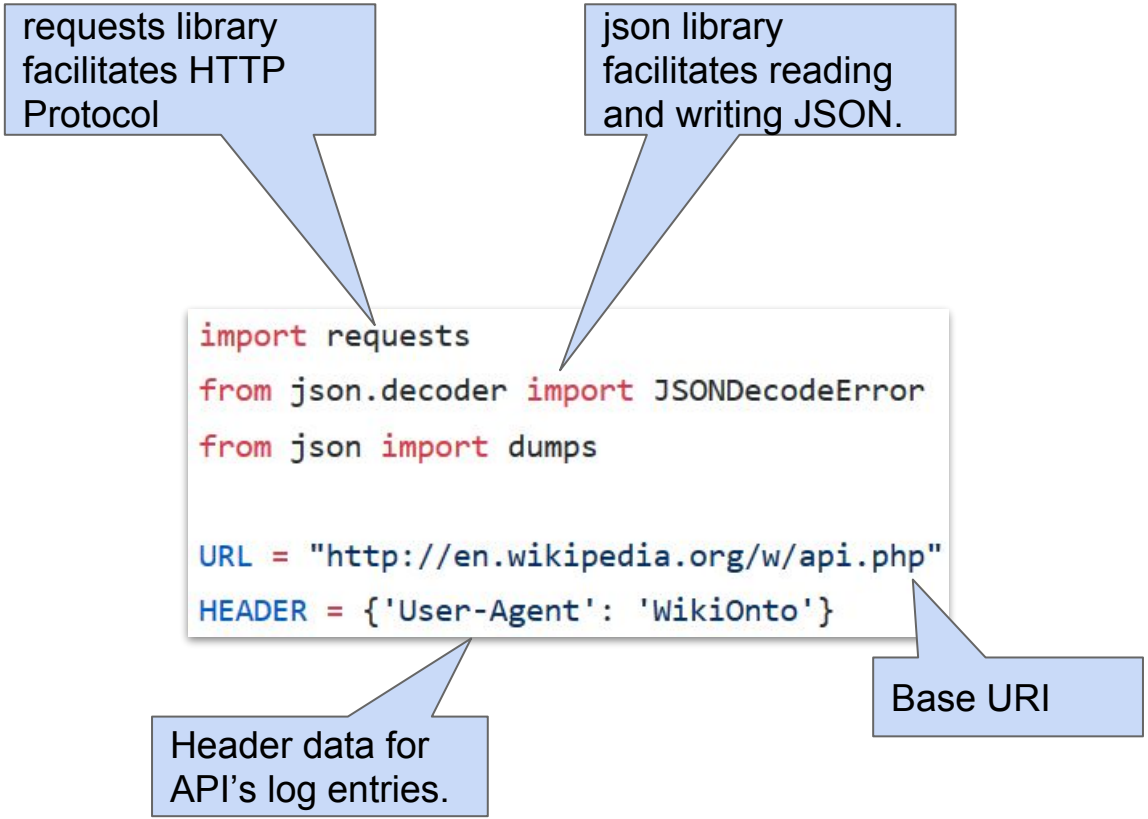
REST – HTTP Messages

See https://en.wikipedia.org/wiki/Representational_state_transfer

Uniform Resource Locator (URL)	GET	PUT	PATCH	POST	DELETE
Element, such as `https://api.example.com/resources/item17`	**Retrieve** a representation of the addressed member of the collection, expressed in an appropriate Internet media type.	**Replace** the addressed member of the collection, or if it does not exist, **create** it.	**Update** the addressed member of the collection.	Not generally used. Treat the addressed member as a collection in its own right and **create** a new entry within it.[17]	**Delete** the addressed member of the collection.

Accessing the API

- Python Requests



Accessing the API

- Python Requests

```
def wiki_request(params):
    params['format'] = 'json'
    params['action'] = 'query'
    params['utf8'] = ''
    try:
        r = requests.get(URL, params=params, headers=HEADER).json()
    except requests.ConnectionError:
        print("Connection Error")
        r = wiki_request(params)
    except JSONDecodeError:
        return None
    return r
```

JSON data is translated to a dictionary and vice versa.

URL parameters.

Only a stack overflow can kick us out. Connection errors can happen. We need fault tolerance.

Accessing the API

- Text content

```
def getcontent(revid):
    params = {'prop': 'revisions'
        , 'rvprop': 'content'
        , 'revids': revid}
    wikijson = wiki_request(params)
    if wikijson is None:
        return None
    try:
        return dumps(wikijson)
    except KeyError:
        return None
```

We need a revision ID for accessing full content

URL parameters.

Return the JSON String representation of the response.

Accessing the API

- Categories

```
def getcategories(title):
    params = {'prop': 'categories'
        , 'titles': title
        , 'redirects': '1'}
    wikijson = wiki_request(params)
    members = next(iter(wikijson["query"]["pages"].values()))["categories"]
    categories = list(map(lambda d: d["title"].replace(" ", "_"), members))
    return categories
```

We only need the title of an article for retrieving its categories.

Accessing the API

- Infobox

```
def get_infobox(pair):
    l = pair[0]
    rev = pair[1]
    text = getcontent(rev).lower()
    if '{{infobox' not in text:
        return l, rev, []
    parts = text.split('{{infobox')
    ibs = []
    for x in range(1, len(parts)):
        p = parts[x]
        name = p.split('|')[0].replace('\\n', '').strip()
        ibs.append(name)
    return l, rev, ibs
```

l is the title.
rev is the Revision ID

We retrieve the name of the template in a very **naive** way.

l = title
r = revision ID
ibs = list of template names with ‘infobox’

Parsing issues

https://www.academia.edu/2861784/MediaWiki_Grammar_Recovery

- There is not always consistency in articles' and categories' titles.
- There is no consistency in the manner the infobox template appears in the Wiki Markup.
  - '{{ infobox programming language' vs '{{Infobox programming language'
- Wiki Markup does not follow a context-free grammar. The language is **context sensitive**.
  - Several transformations (~20 transformations exist) can be applied such as resolving templates to retrieve a context-free language.
  - There is no parser that completely implements Wiki Markup.
  - Parsoid translates Wiki Markup to HTML, but does not seem to have a proper intermediate AST representation.

Limitations

- You should always try to keep the number of requests to the Web API low.
- The API may exclude your IP from access, if you cause too much traffic. Then, you have to wait.
- Every time, you send a HTTP request, you need to wait for a response. This heavily affects tool performance.
  - You can start multiple processes sending HTTP requests and waiting for responses.
  - If you need to send > 1k requests, you should consider downloading a data dump or accessing Wikipedia through DBpedia.

DBpedia

http://live.dbpedia.org/page/Java_(programming_language)

- Wikipedia's data is preprocessed and translated into RDF data.
  - This means everything is saved as subject property object.
  - Read more on RDF at https://www.w3.org/RDF/
- Every page is treated as a resource with its link being its unique identifier. (e.g., **http://dbpedia.org/resource/Java_(programming_language)**)
- A Wikipedia article's summary is associated through the property dbo:abstract.
- The original Wikipedia article link from which the data was extracted is associated through the property dbo:wikiPageRevisionLink.
- The infobox template being used is associated through dbp:wikiPageUsesTemplate.

Accessing DBpedia

- Requests

- You can send requests to the SPARQL endpoint similar to the access to Wikipedia.
  - See https://github.com/softlang/wikionto/blob/master/src/mine/dbpedia.py for example code.
  - Endpoint URL http://dbpedia-live.openlinksw.com/sparql

Accessing DBpedia

- Categories

```
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbr_cat: <http://dbpedia.org/resource/Category:>
SELECT ?article where {
  SELECT DISTINCT ?article where {
    ?article dct:subject/skos:broader{0,5} dbr_cat:Programming_languages.
  }
  ORDER BY ASC(?article)
}
limit 10000
offset 0
```

DBpedia
- Limitations

- Depending on the complexity of the query, the SPARQL engine may go **out of memory**.
  - Retrieving articles to lower depths is hard.
  - If the data is split into more packets, the number of requests is raised.
- Infobox template names are collected.
- Not every Infobox property is collected. Only some are matched to attributes in DBpedia, e.g. dbp:paradigm
- Not every Infobox property value is collected properly. Values are not normalized.
  - See paradigm of Java: “Multi-paradigm: Object-oriented , structured, imperative, generic, reflective, concurrent”

Wikipedia – dumps

- Download dumps here https://dumps.wikimedia.org/enwiki/
  - ...page.sql for mapping article names to their internal IDs. (~5GB)
  - ...categorylinks.sql for extracting category membership. (~18GB)
  - ...page-articles-multistream.xml for page content. (~67GB)
  - ...pagelinks.sql (~50GB)
- When processing article content from the XML, you have to use streaming parsers. No ASTs can be build in memory.
- When processing SQL dumps, you should set up a database and load the data from the dumps.
- English Wikipedia has 5,762,091 articles.

- Processing dumps is a non-trivial task.
  - You need hardware resources.
  - You need time (even if you parallelize).
  - You need the right technologies, such as Python lxml.

Summary

- Access small amounts of information from Wikipedia through the REST API.
- Access small to medium amounts of structured information from Wikipedia through DBpedia.
- Access large amounts of information from dumps.
- Watch out for the need of data preprocessing.
- Performance plays a huge role for data processing.
- More on technologies enabling performance improve soon.