Project Module

Natural Language Processing and Semantic Technologies

Dr. Sebastian Furth
University of Applied Sciences Würzburg – Schweinfurt
Summer Term 2022

Part 1: Construction of a parallel, multilingual corpus

Project Module – Natural Language Processing and Semantic Technologies

FH·W·S

What is a corpus?

- A text corpus is a very large collection of text (often many billion words) produced by real users of the language and used to analyze how words, phrases and language in general are used. It is used by linguists, lexicographers, social scientists, humanities, experts in natural language processing and in many other fields.

- A corpus is also be used for generating various language databases used in software development such as predictive keyboards, spell check, grammar correction, text/speech understanding systems, text-to-speech modules, machine translation systems and many others.

Types of Text Corpora

- **Monolingual corpus**
  A monolingual corpus is the most frequent type of corpus. It contains texts in one language only. The corpus is usually tagged for parts of speech and is used by a wide range of users for various tasks from highly practical ones, e.g. checking the correct usage of a word or looking up the most natural word combinations, to scientific use, e.g. identifying frequent patterns or new trends in language.

- **Parallel corpus, multilingual corpus**
  A parallel corpus consists of two or more monolingual corpora. The corpora are the translations of each other. For example, a novel and its translation or a translation memory of a Computer Assisted Translation (CAT) tool could be used to build a parallel corpus. Both languages need to be aligned, i.e. corresponding segments, usually sentences or paragraphs, need to be matched. The user can then search for all examples of a word or phrase in one language and the results will be displayed together with the corresponding sentences in the other language. The user can then observe how the search word or phrase is translated.

Tasks

1. Crawl technical documents of a certain domain from the internet

- First choose your domain of interest.
- **Tipp:** Coffee Machines could be a good fit, as many vendors offer technical documents in multiple languages.

- **Requirements**

  - The corpus must comprise documents from different vendors with multiple languages each
  - This must be reproducable and extensible in the future!
  - It is not sufficient to submit a collected data set of manually downloaded documents!
  - Create a configurable/extensible index of websites / URLs that shall be crawled
    (i.e. define page where the crawler starts looking for downloads)
  - Create an (intelligent) script for each website / vendor that downloads the respective document
    (i.e. do not define all URLs manually, instead try to have some „rules“ (e.g. based on HTML/CSS information) that can identify hyperlinks to download the documents)
  - If the script is run again, only new documents shall be downloaded and stored.

Tasks

2. Preprocess Documents

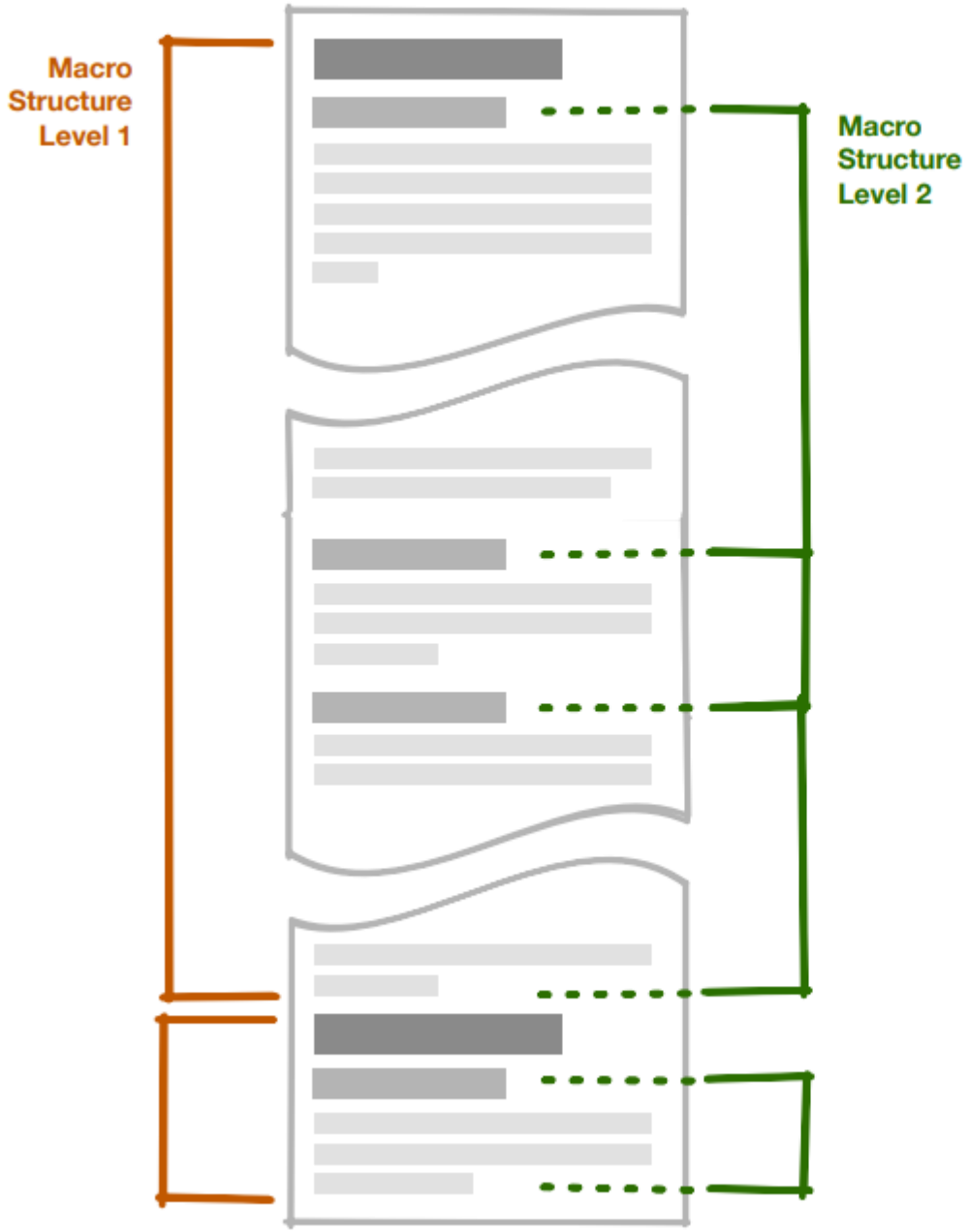
- Since the downloaded documents are usually only available as PDFs, pre-processing must also be integrated in your project, i.e. extraction and segmentation of the text.
- The main challenge for building up a text corpus from PDF documents is to accurately export the text of single segments from the PDF document.

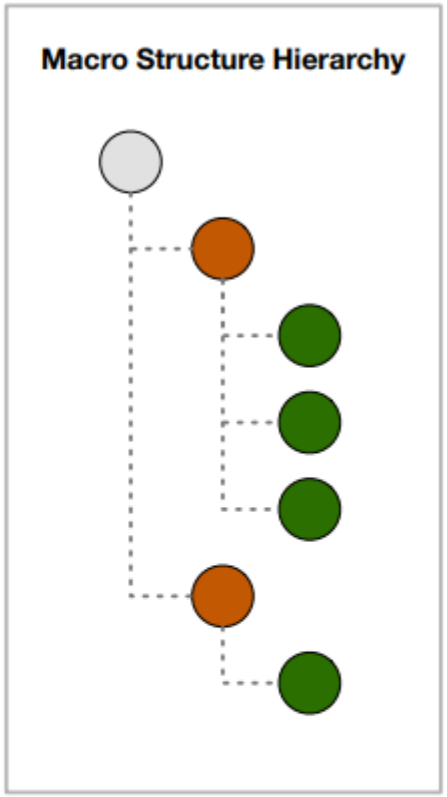
- **Requirements**
  - The text of the complete document should be hierarchically segmented, i.e. chapters/sections/sub-sections etc. – alternatively, you can choose a reasonable segmentation level (e.g. all sections)
  - Again, the complete process must be reproducible.
  - Submitting a collection of preprocessed texts is not sufficient!

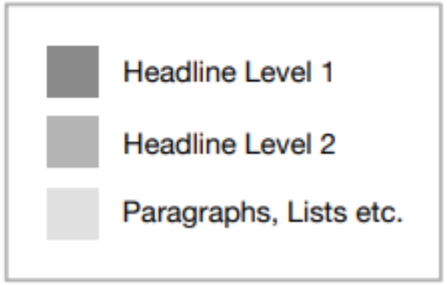
- **Tipps**
  - You can check, if available libraries have sufficient results (e.g. https://github.com/MBAigner/PDFSegmenter)
  - If not, you must work with the low-level PDF blocks on your own (hint: headlines are key here)
    - A possibility to identify headlines is to exploit the PDF bookmarks.
    - Some libraries (like pdf2xml - https://github.com/kermitt2/pdf2xml) can export the table of contents (PDF bookmarks) from the document as XML.

Tasks

2. Preprocess Documents







Tasks

2. Preprocess Documents

Algorithm 1 Recursive algorithm for Macro Structure Recovery.

```
1:  // Start recovery: Create root macro structure for complete document
2:  root ← MacroStructure.new
3:  root.begin ← document.begin
4:  root.end ← document.end
5:  root.level ← 0
6:  CREATEMACROSTRUCTURES(root, 0)
7:
8:  // Recursively create macro structures on different levels
9:  function CREATEMACROSTRUCTURES(parent : MacroStructure, level : int)
10:     headlines ← GETCOVEREDMICROSTRUCTURES(parent, HEADLINE, level)
11:     while HASNEXT(headlines) do
12:         headline ← NEXT(headlines)
13:         current ← CREATEMACROSTRUCTURE(headline)
14:         if previous is not null then
15:             previous.end ← current.begin − 1
16:             CREATEMACROSTRUCTURES(previous, level + 1)
17:         end if
18:         if not HASNEXT(headlines) then
19:             current.end ← parent.end
20:             CREATEMACROSTRUCTURES(current, level + 1)
21:         end if
22:         previous ← current
23:     end while
24: end function
25:
26: // Actually create a single structure
27: function CREATEMACROSTRUCTURE(headline : MicroStructure)
28:     structure ← MacroStructure.new
29:     structure.begin ← headline.begin
30:     structure.level ← headline.level
31:     return structure
32: end function
```


FH·W·S

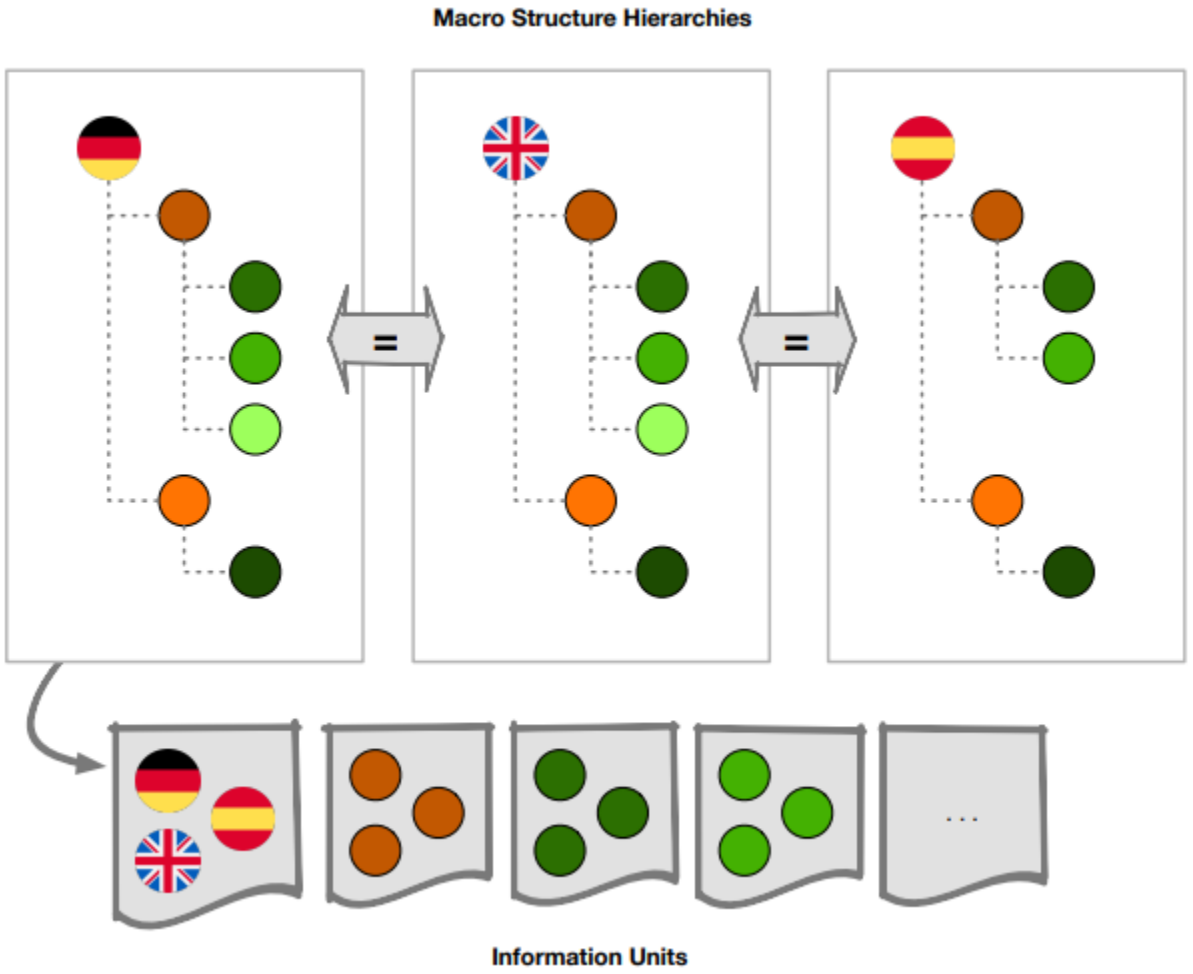
Tasks

3. Align segments of documents that are available in mulitple languages

- In a parallel, multilingual corpus, both languages need to be aligned, i.e., corresponding segments, usually sections or paragraphs, need to be matched.
- **Requirements**
  - If for a document multiple languages exist, their segments need to be aligned
- **Tipps**
  - For this project a simple mechanism, e.g. based on chapter numbers is sufficient
  - Sometimes PDF documents contain multiple languages. This must be handled accordingly. You can use libraries like langdetect for that (https://github.com/Mimino666/langdetect)

Tasks

3. Align segments of documents that are available in mulitple languages



Tasks

4. Output Format

- In the end, a corpus must be easily accessible through NLTK and Hugging Face
- **Requirements**
  - Please check carefully the respective data formats of NLTK and Hugging Face
  - Your corpus must be (easily) accessible through the respective APIs/Functions of NLTK and Hugging Face
  - Please make sure to provide respective scripts that demonstrate loading the data using NLTK and Hugging Face
- **Tipps**
  - Hugging Face: https://huggingface.co/docs/datasets/loading#local-and-remote-files
  - NLTK: https://www.nltk.org/howto/corpus.html

Part 2: Building a technical ontology

Project Module – Natural Language Processing and Semantic Technologies

Tasks

4. Build a semantic representation of your domain (e.g. coffee machine)

A. **Power button**
Button light flashes while machine is heating.

B. **Integrated & removable 54mm tamper**

C. **Group head**
For easy positioning of the portafilter.

D. **54mm stainless steel portafilter**
With commercial style spouts.

E. **Extra-tall cup clearance**
For tall mugs.

F. **Removable drip tray**
With Empty Me! tray full indicator.

G. **Storage tray** (located behind drip tray)
Houses accessories when not in use.

H. **360° swivel action steam wand**
Adjusts to the perfect position for texturing.

I. **Dedicated hot water outlet**
Delivers instant hot water for Americanos & pre-heating cups.

J. **Steam/Hot Water dial**

K. **1 CUP and 2 CUP buttons**
With preset, manual over-ride or reprogrammable shot volumes.

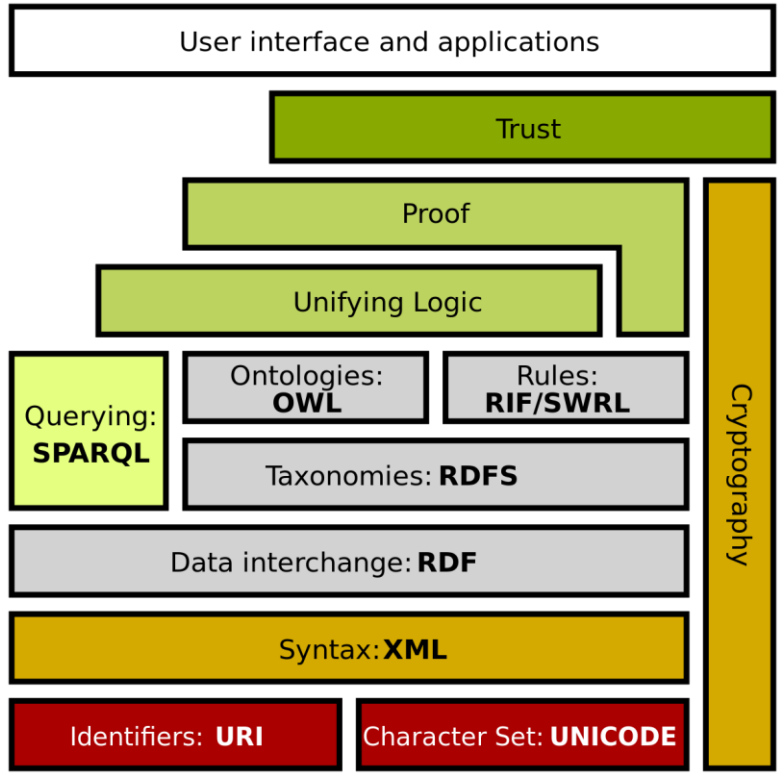
L. **Steam/Hot Water light**
Illuminates to indicate that the steam/hot water function is selected.

M. **CLEAN ME light**
Illuminates when a cleaning cycle is required.

Tasks

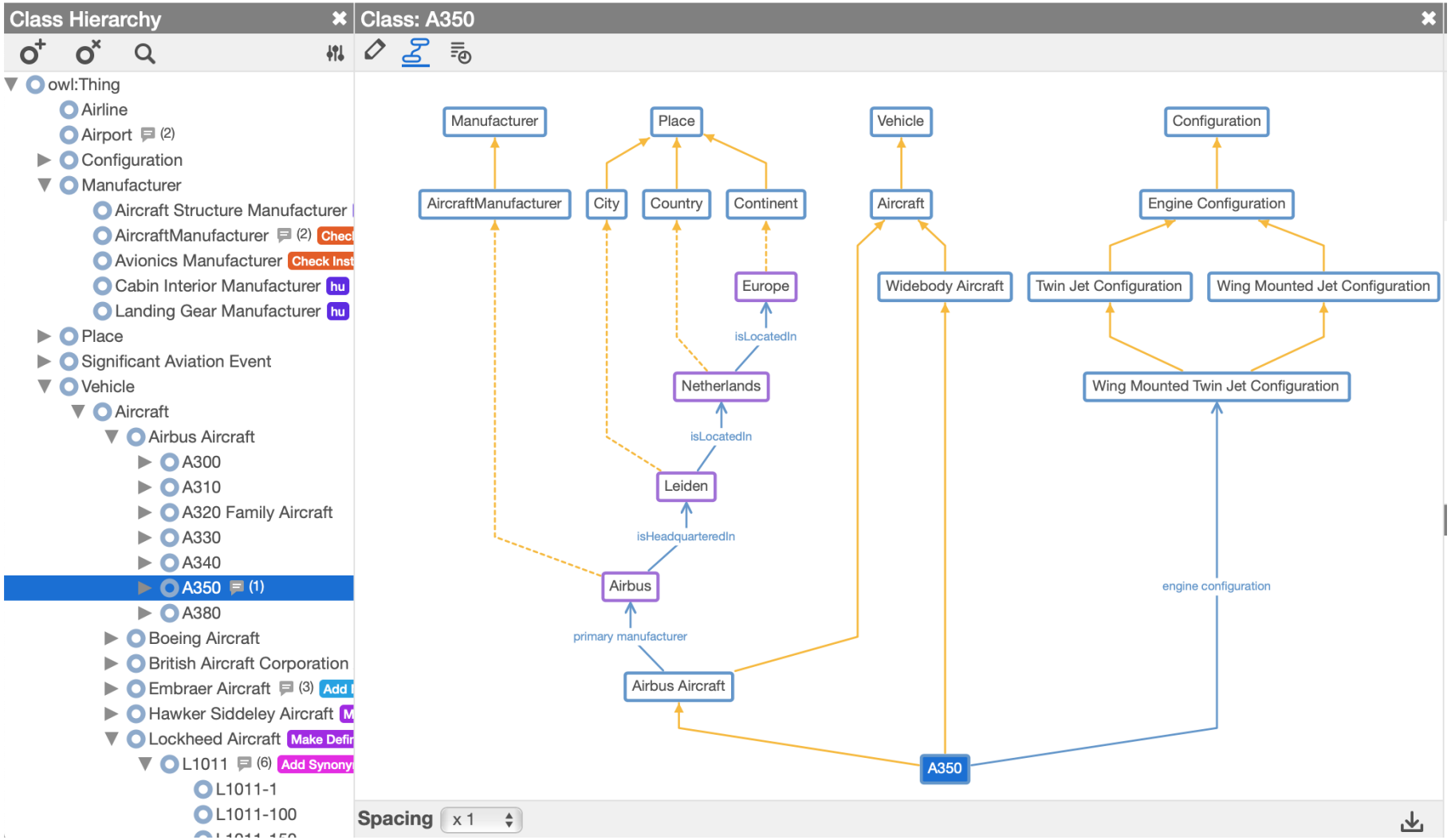
4. Build a semantic representation of your domain (e.g. coffee machine)

- Use the Semantic Web Technologies that are standardized by W3C, especially RDF.
- **Requirements**
  - The semantic representation must describe at least
    - Common components of your machine in focus
    - Common functions of your machine in focus
  - Please have a look at some of your documents and build a super-set of the respective items
  - Please work with classes, relationships and instances!
- **Tipps**
  - This is not rocket science, but a straightforward task
  - You don't need to become an expert on Semantic Web technologies!
  - You can use WebProtege for working collaboratively on your ontology (https://webprotege.stanford.edu/)



Tasks

4. Build a semantic representation of your domain (e.g. coffee machine)



Example ontology on Aircrafts in WebProtege

Part 3: Text Analytics (Winter Term 2022-2023)

Project Module – Natural Language Processing and Semantic Technologies

Preview

- You will use your prepared corpus and the ontology to do advanced text analytics.

  - **Text Classification**
    - The concepts of your ontology (e.g. components of the coffee machine) will be used as topics (classes)
    - Advanced Deep Learning techniques for NLP will be used to classify the prepared segments in your corpus according to these topics

  - **Question Answering**
    - Question Answering comes in many flavours, but the one we'll focus is called extractive question answering. This involves posing questions about a document and identifying the answers as spans of text in the document itself. Therefore, we will reuse your prepared corpus.

  - Other options are **Automatic Summarization**, **Translation**, …

Organizational Aspects

FH·W-S

Organizational Aspects

- Contact: Dr. Sebastian Furth (sebastian.furth@lehrbeauftragte.fhws.de)

- (Urgent/Blocking) Questions on the project can be sent at any time via mail.

- We can have „checkpoint“ meetings to check your progress, gather feedback, and to discuss open questions.

  - You are the main actors in these meetings!
  - I will give feedback and answer your questions.
  - Please be prepared for these meetings and have your progress ready to be shared.
    (Short demos of what you have already achieved are highly appreciated)

- For your code, please create a GitHub repository and invite me as collaborator (*@sebastianfurth*)

Checkpoint Meetings (Part 1 + Part 2 – Summer Term 2022)

- **Checkpoint 1**
  - Domain chosen
  - List of vendors prepared
  - Downloadability of documents checked
- **Checkpoint 2**
  - Download script for one vendor implemented and tested
  - Preprocessing script for one vendor implemented and tested
- **Checkpoint 3**
  - Ontology created
- **Final Submission**
  - Corpus collected with documents from all vendors
  - Corpus accessible via Hugging Face and NLTK

Recommended Readings

- **Foundations of Semantic Web Technologies**
  - Pascal Hitzler, Sebastian Rudolph, Markus Krötzsch
  - Chapman & Hall, 2009

- **Natural Language Processing with Transformers: Building Language Applications with Hugging Face**
  - Lewis Tunstall, Leandro von Werra, Thomas Wolf
  - O'Reilly, 2022