# Teaching *"Unstructured Information Management: Theory and Applications"* to Computational Linguistics Students

Iryna Gurevych, Christof Müller, Torsten Zesch

Ubiquitous Knowledge Processing Group
Telecooperation, Computer Science Department
Darmstadt University of Technology

# Typical NLP course

- Project topic
  - Yet another tokenizer
- Project results
  - Unstable software
  - Works only under special preconditions
  - Hard-coded configuration
    - “The software has to be installed in directory *`foo`*“
    - “The name of the input file has to be *`foobar`*”

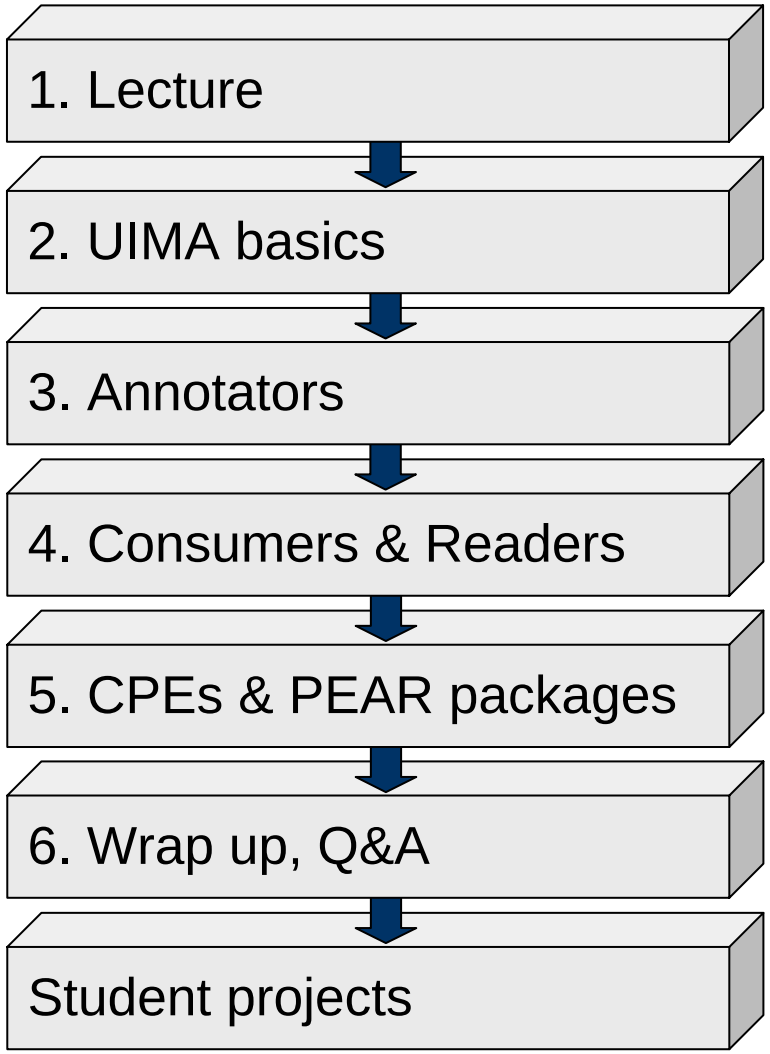
# Goals of our NLP course

- Teach basics in unstructured information management
- Separate software engineering from NLP
  - Provide a framework and preprocessing components
- Enabling students to:
  - Concentrate on computational linguistics part
  - Work on more challenging/motivating tasks

Using UIMA to reach these goals

# Course outline

- Compact seminar
  - 6 sessions
  - 4 hours each
- Course requirements (MA level)
  - Participation
  - Implement a practical project
  - Deliver results as PEAR package
  - Write a course paper

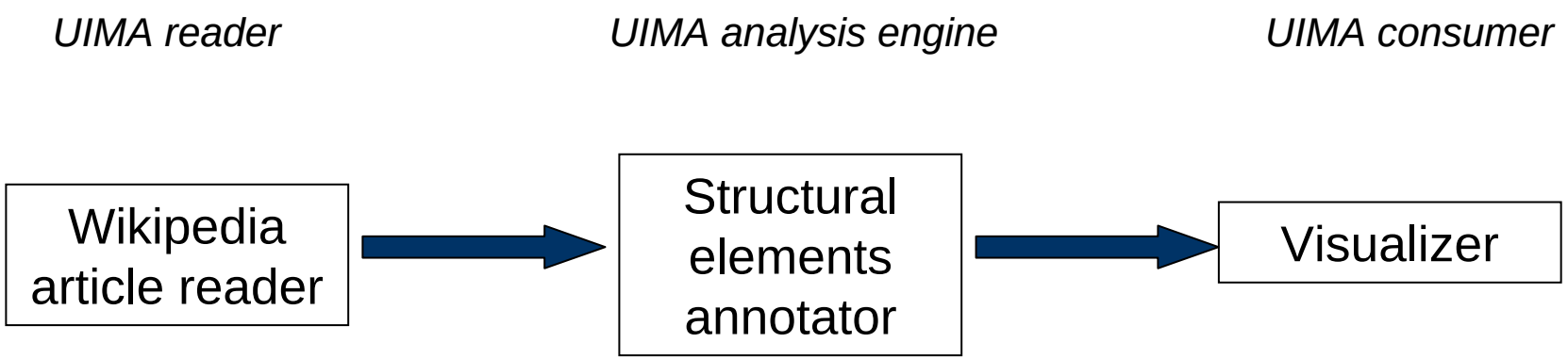


# Student projects

- Suitable task were defined in collaboration with lecturers
- Selected projects:
  - Annotating Wikipedia articles
  - Extracting lexical semantic information from blogs
  - Named entity recognition
  - Sentiment detection
  - Word sense disambiguation

# Annotating Wikipedia Articles

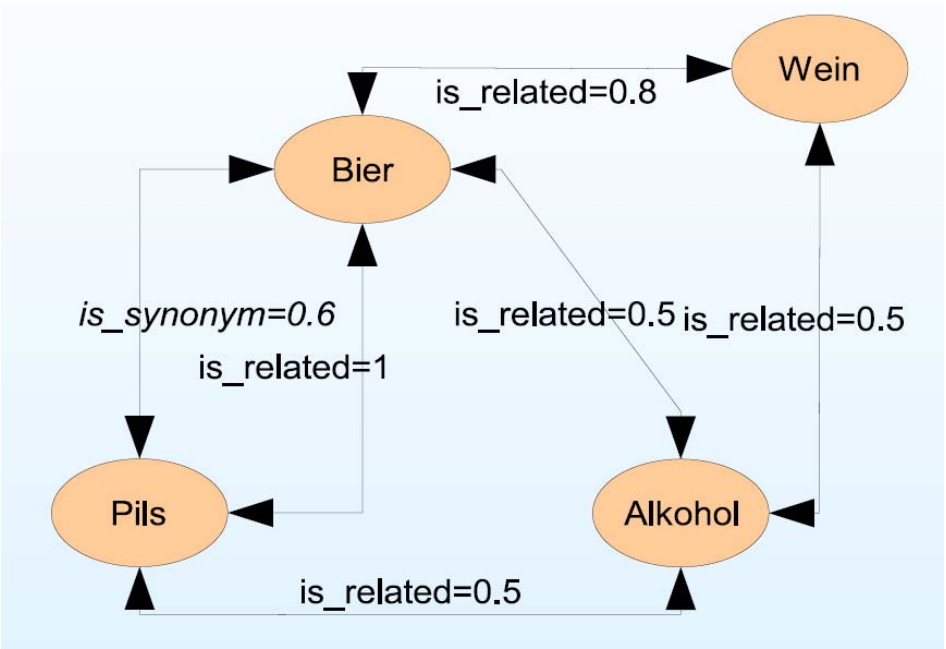
- Annotate structural elements in Wikipedia articles
  - Sections, paragraphs, lists, bold terms, ...
- Visualize annotations
- Wikipedia API is provided to retrieve articles



# Lexical Semantic Information from Blogs

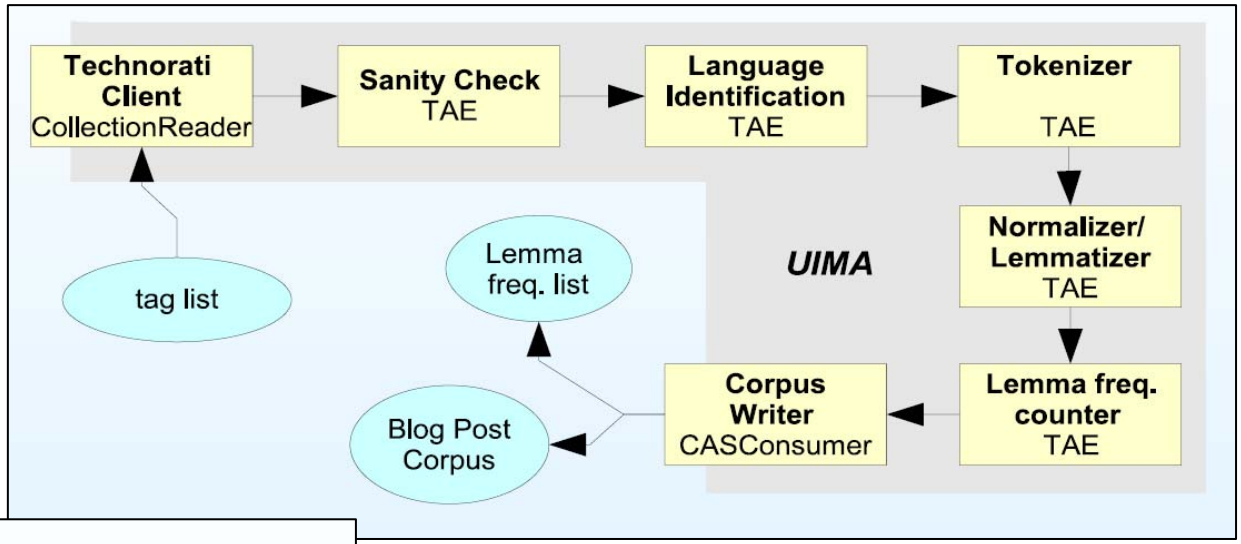
- Analyze blogs
- Find keywords
- Detect semantic relations between keywords

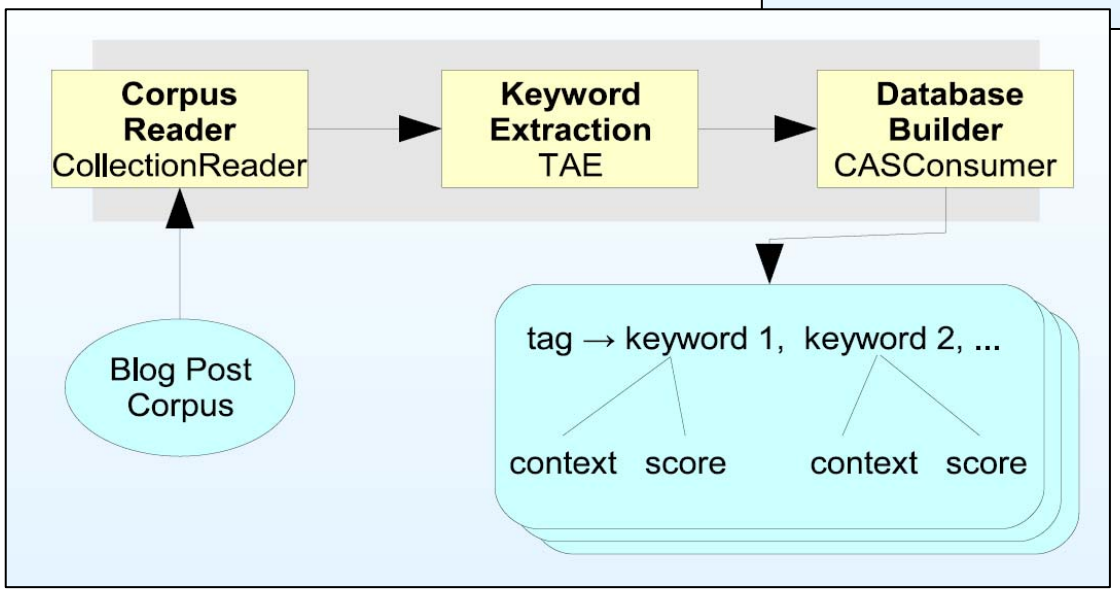
Desired output:



# Lexical Semantic Information from Blogs

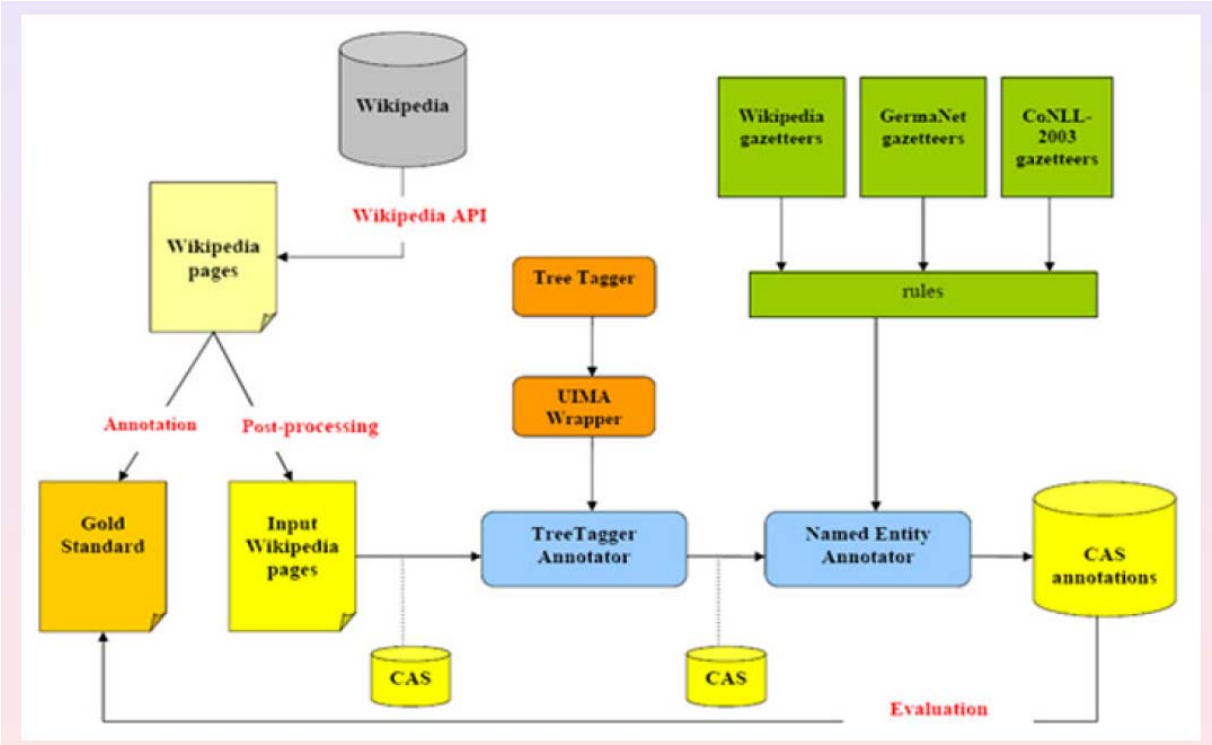
UIMA components as proposed by the students.



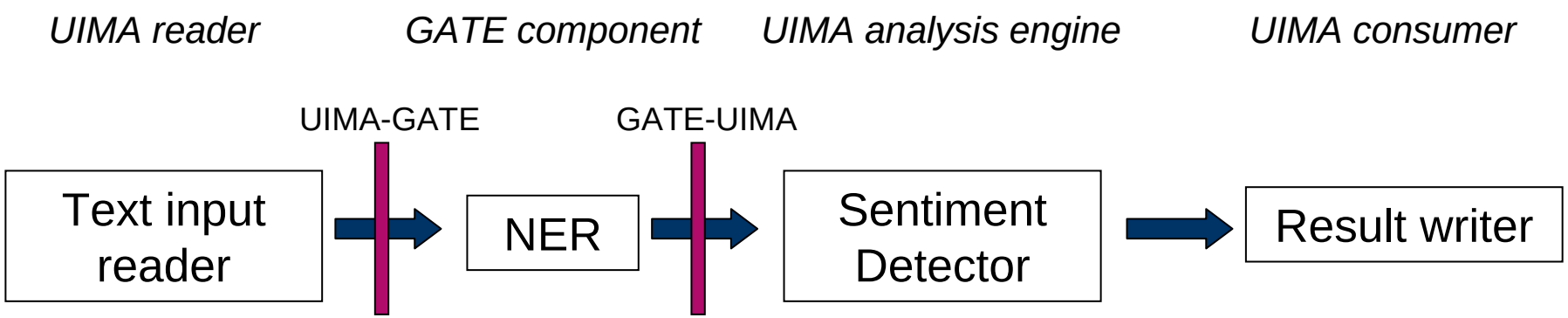
# Named Entity Recognition

- Hybrid approach: rules + gazetteers
- Preprocessing components were provided
- GermaNet and Wikipedia are accessed as UIMA resources



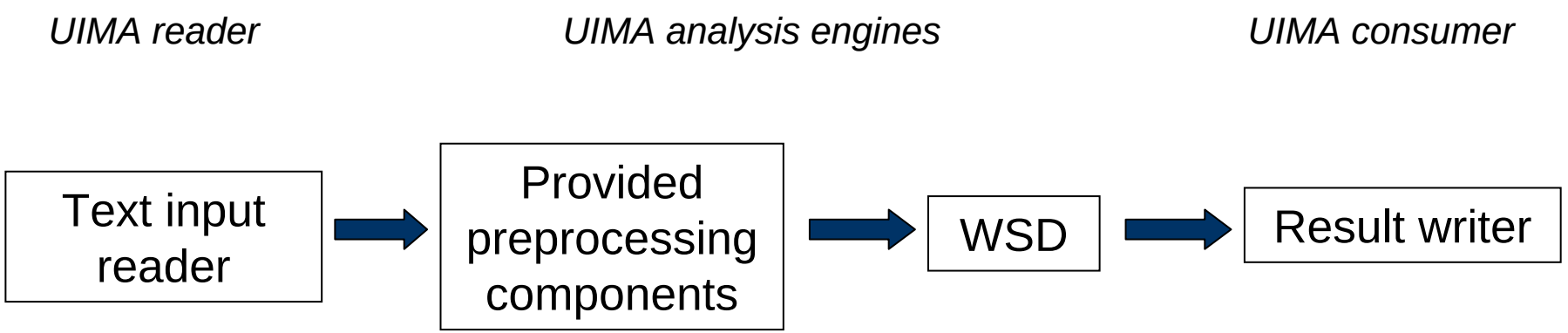
# Sentiment Detection

- Detect sentiment expressions and link them with the judged entity
- Preprocessing components were provided
- Robust NER component is required, but not yet available for UIMA
- Used GATE-UIMA interoperability layer to integrate ANNIE tool



# Word Sense Disambiguation

- Implements the WSD approach by Patwardhan and Pedersen (2006)
- Necessary word glosses are generated using GermaNet
- GermaNet is accessed as a UIMA resource
- Preprocessing components were provided




# Lessons Learned

- Advantages of using UIMA
  - Provide necessary preprocessing tools
  - Enables more challenging/motivating tasks
  - Uniform structure of project results (PEAR package)
  - Students can concentrate on their core competences
  - Focus is on modeling rather than programming
- Challenges
  - Complexity of UIMA architecture
  - Motivate students
- Possible solution
  - Provide a preconfigured work environment **vs.** Learn UIMA


# Thank you very much!

http://www.ukp.tu-darmstadt.de/


- Acknowledgments:
  - Prof. Erhard Hinrichs for his idea to offer the course
  - ISCL students participating
    - Jonathan Khoo, Niels Ott, Sladjana Pavlovic, Maria Tchalakova, Bela Usabaev, Desislava Zhekova, Ramon Ziai