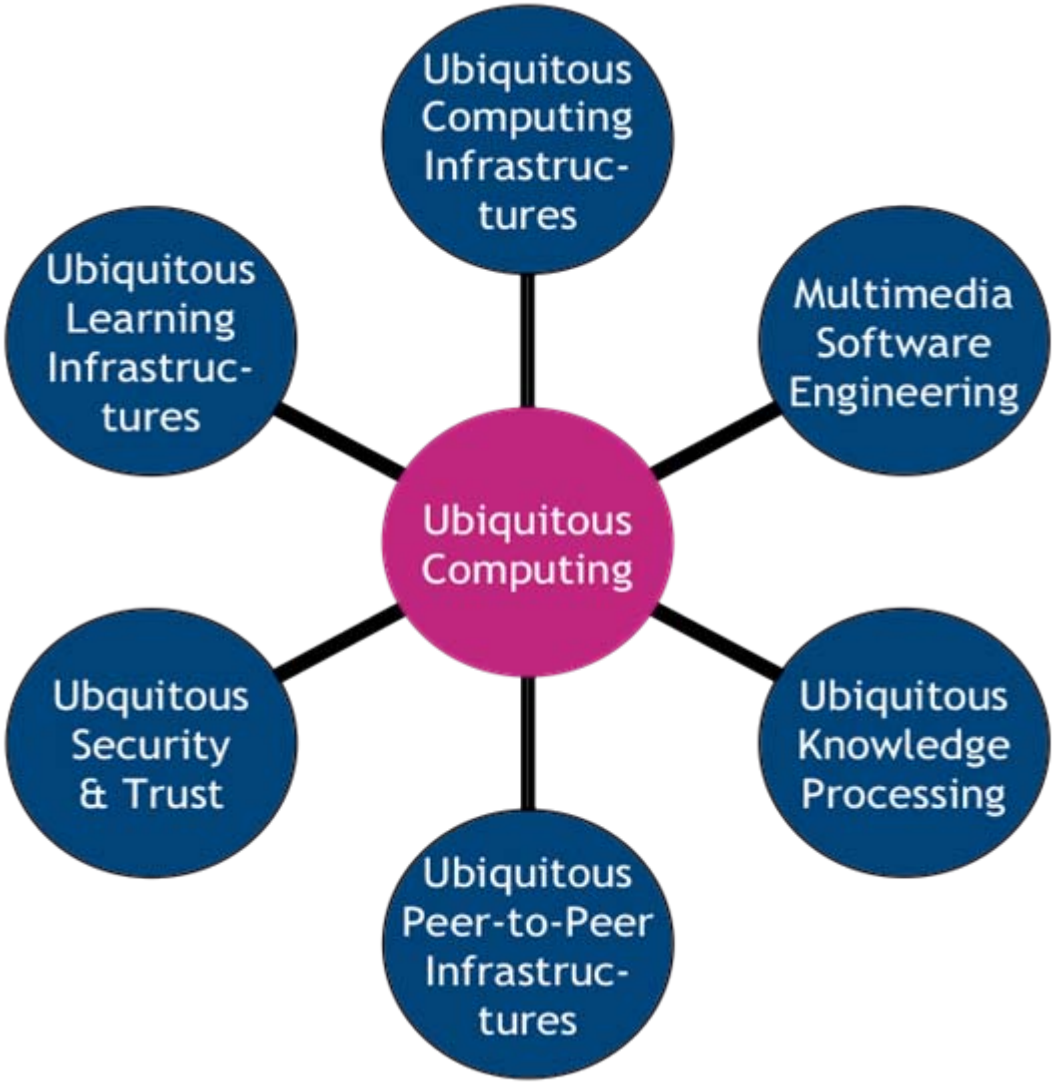
Darmstadt Knowledge Processing Repository Based on UIMA

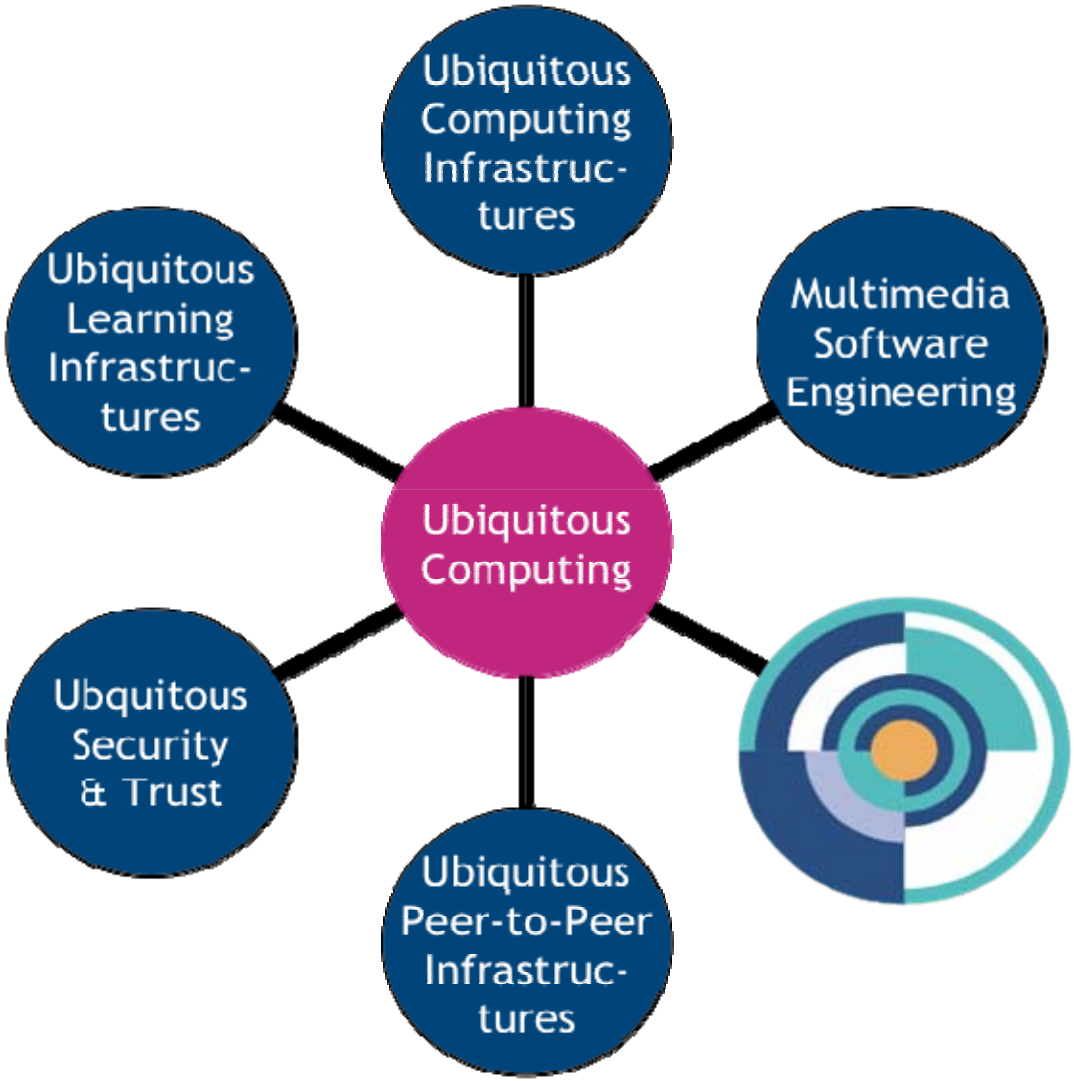
Iryna Gurevych, Max Mühlhäuser, Christof Müller,
Jürgen Steimle, Markus Weimer, Torsten Zesch

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

Telecooperation



Telecooperation



Ubiquitous Knowledge Processing

Ubiquitous Knowledge Processing





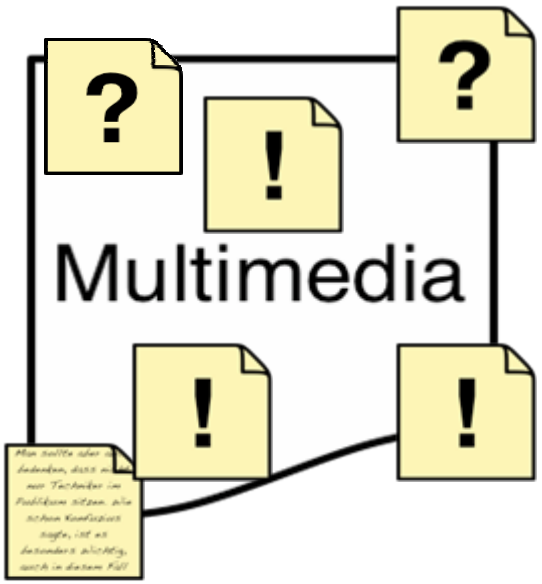
SIR

AQUA

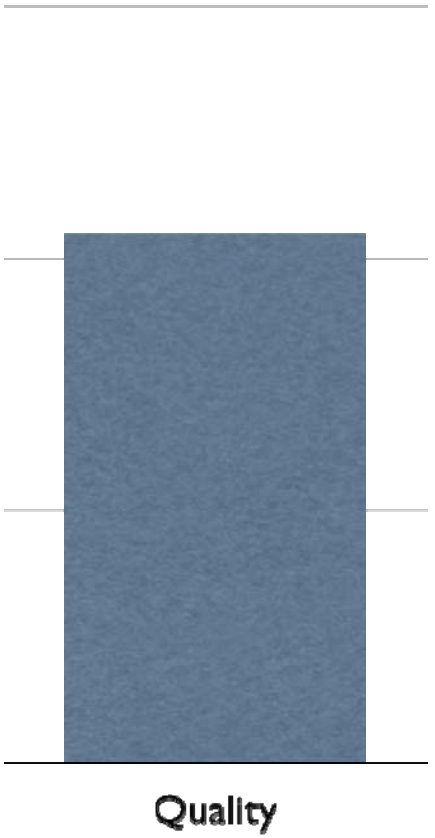
THESEUS

Darmstadt Knowledge Processing Software Repository

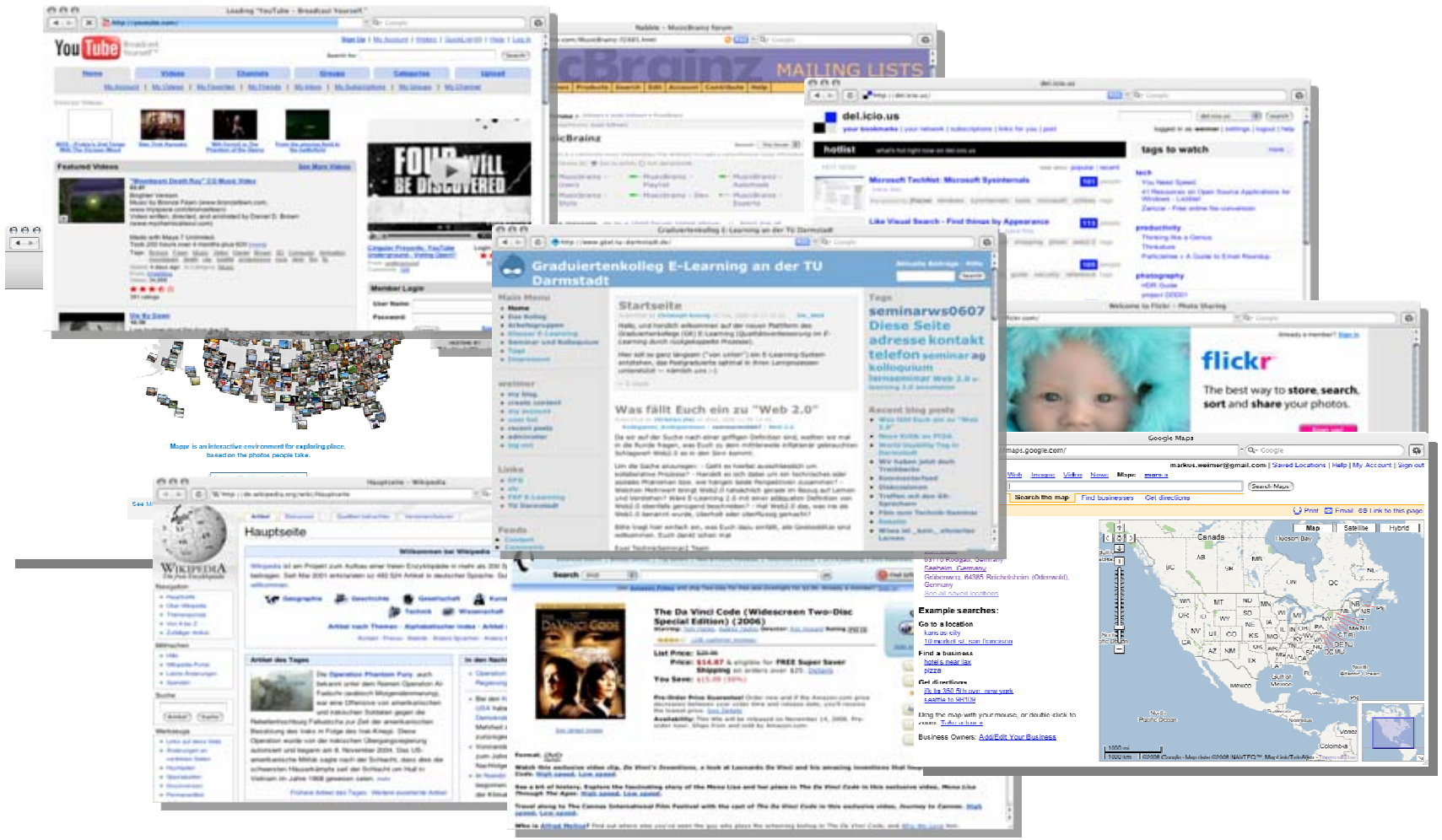
Automatic **Qu**ality **A**ssessment and Feedback in eLearning 2.0 (AQUA)



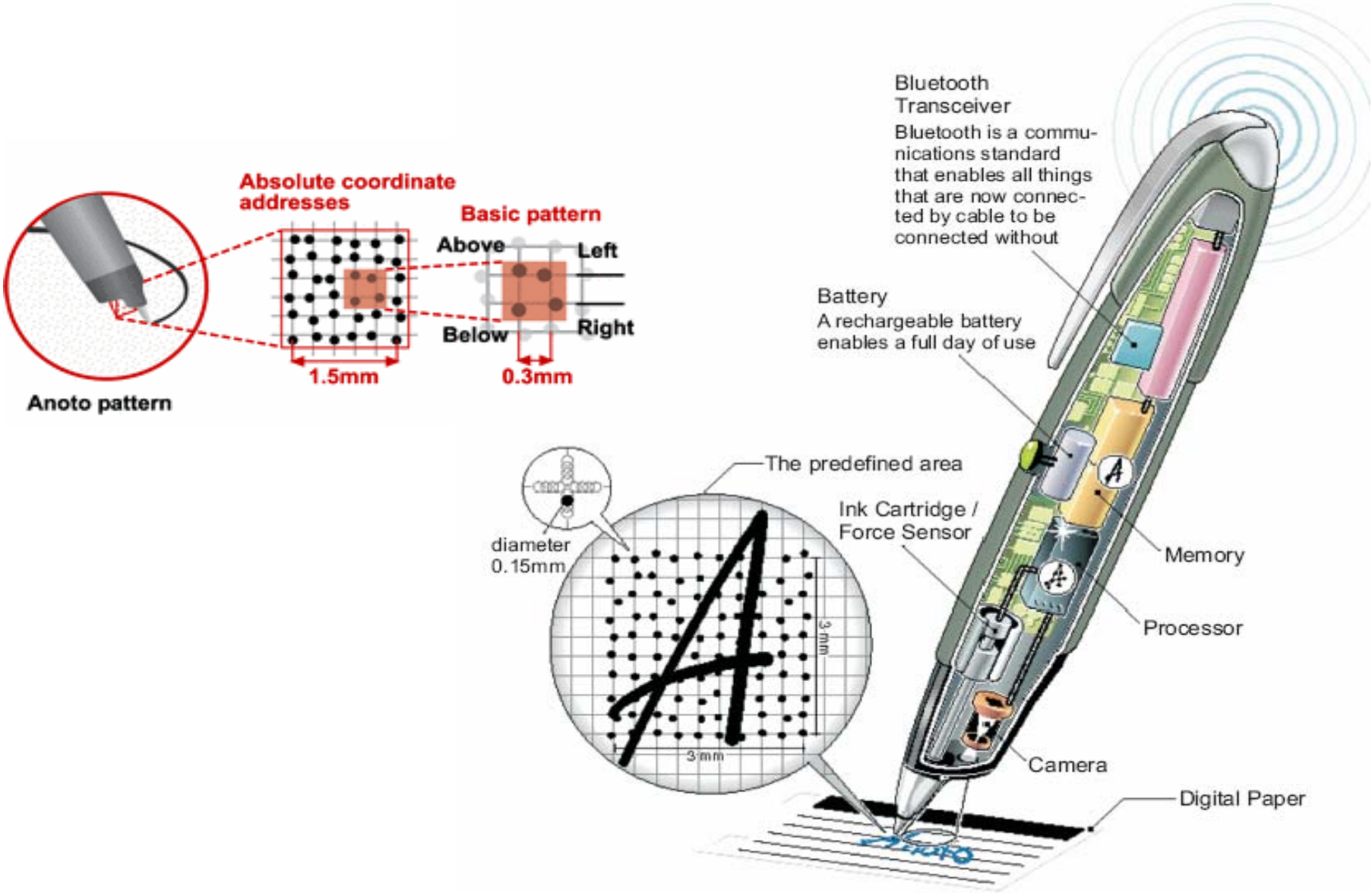
AQUA



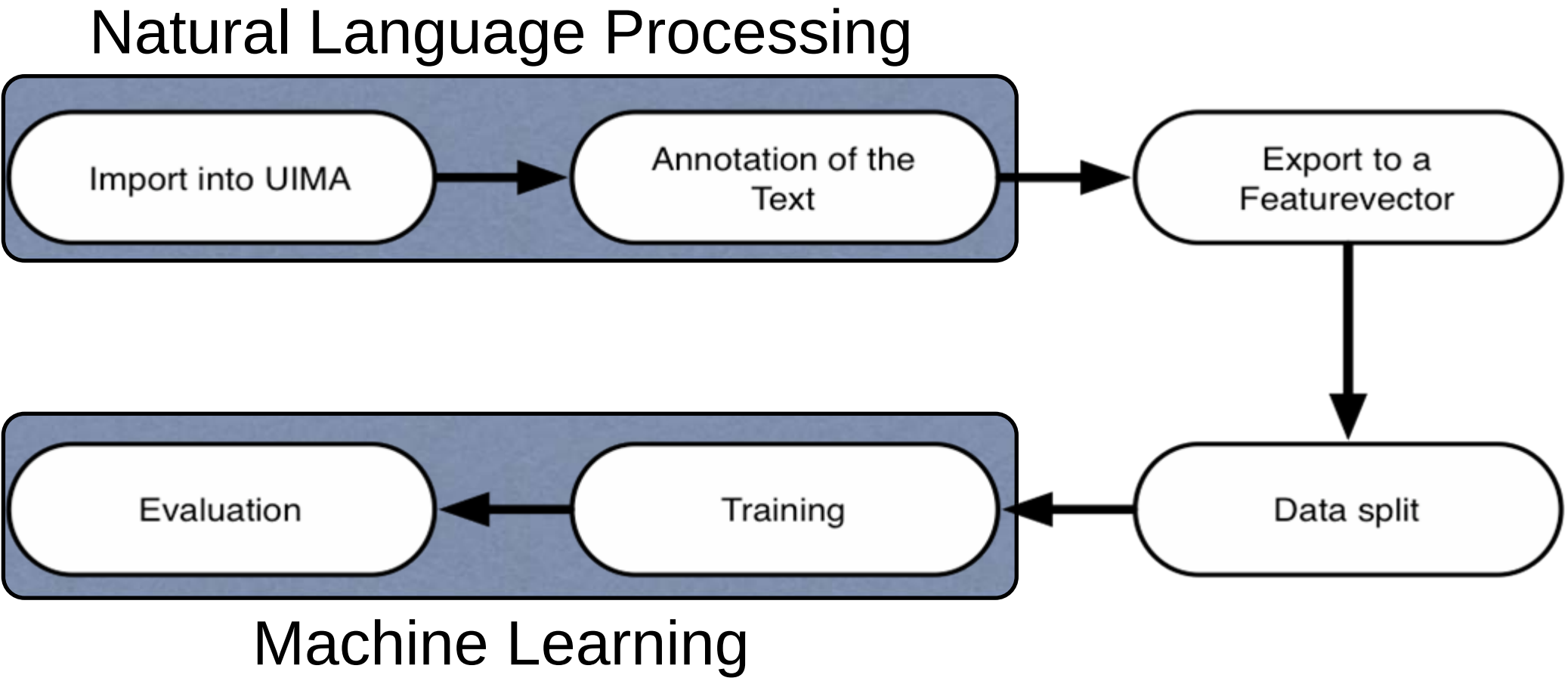
User Generated Discourse in Web 2.0



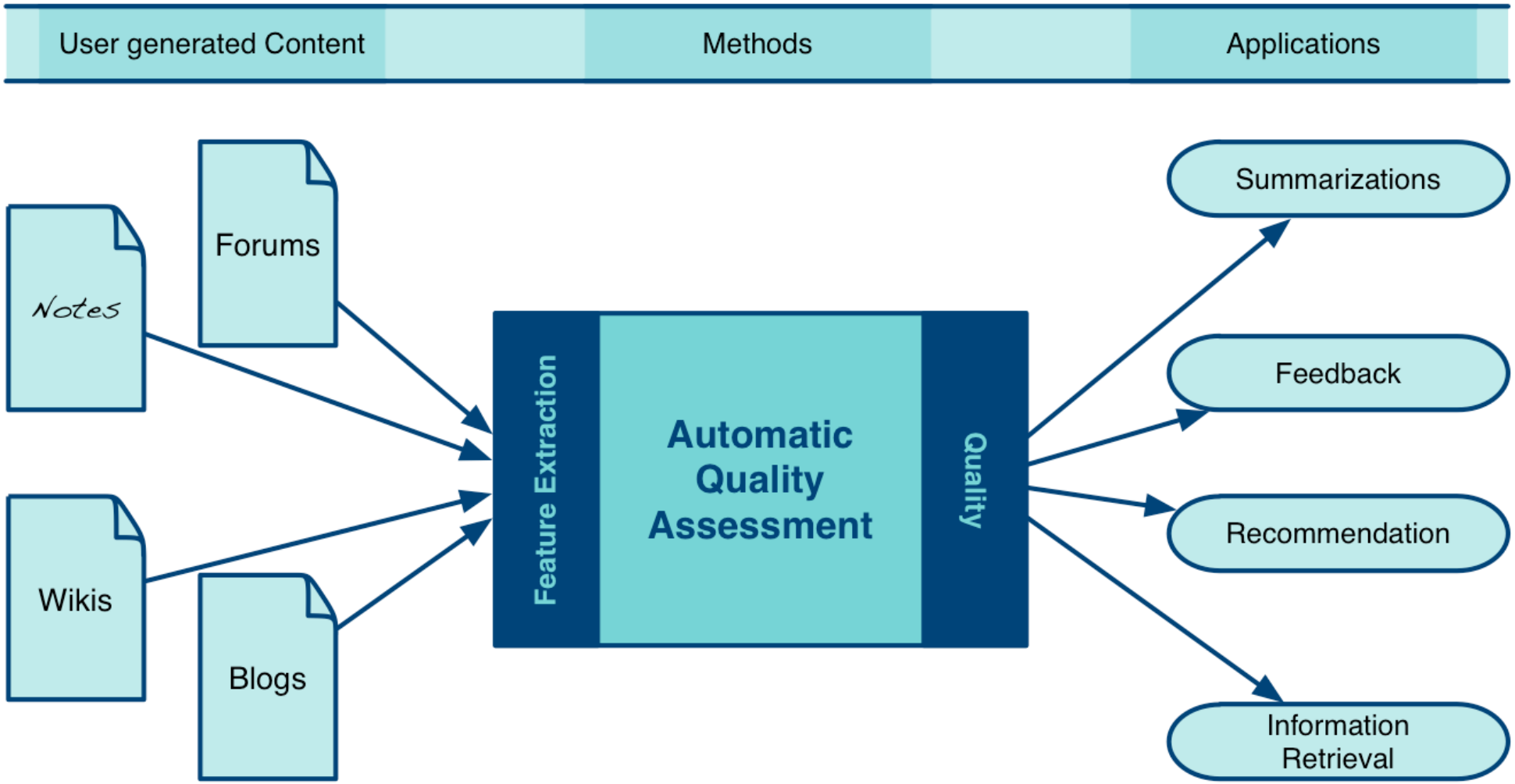
AQUA – Anoto pen



AQUA - System Architecture

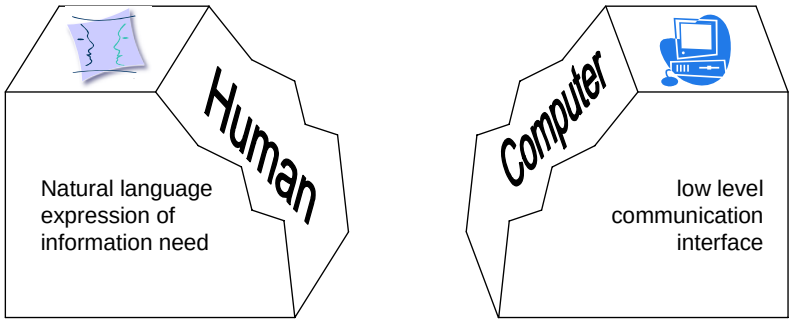


AQUA – System Architecture

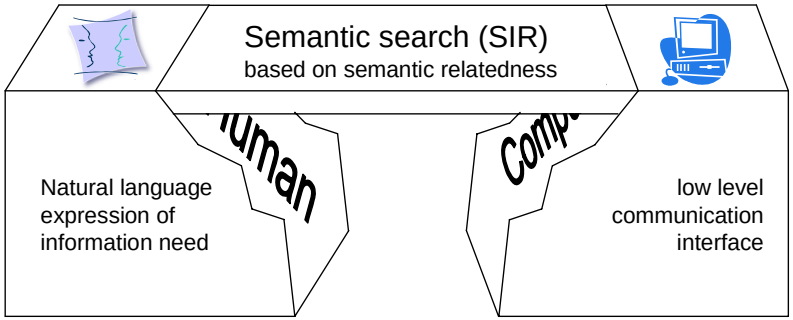


SIR (in cooperation with Prof. Hinrichs)

- Semantic Information Retrieval

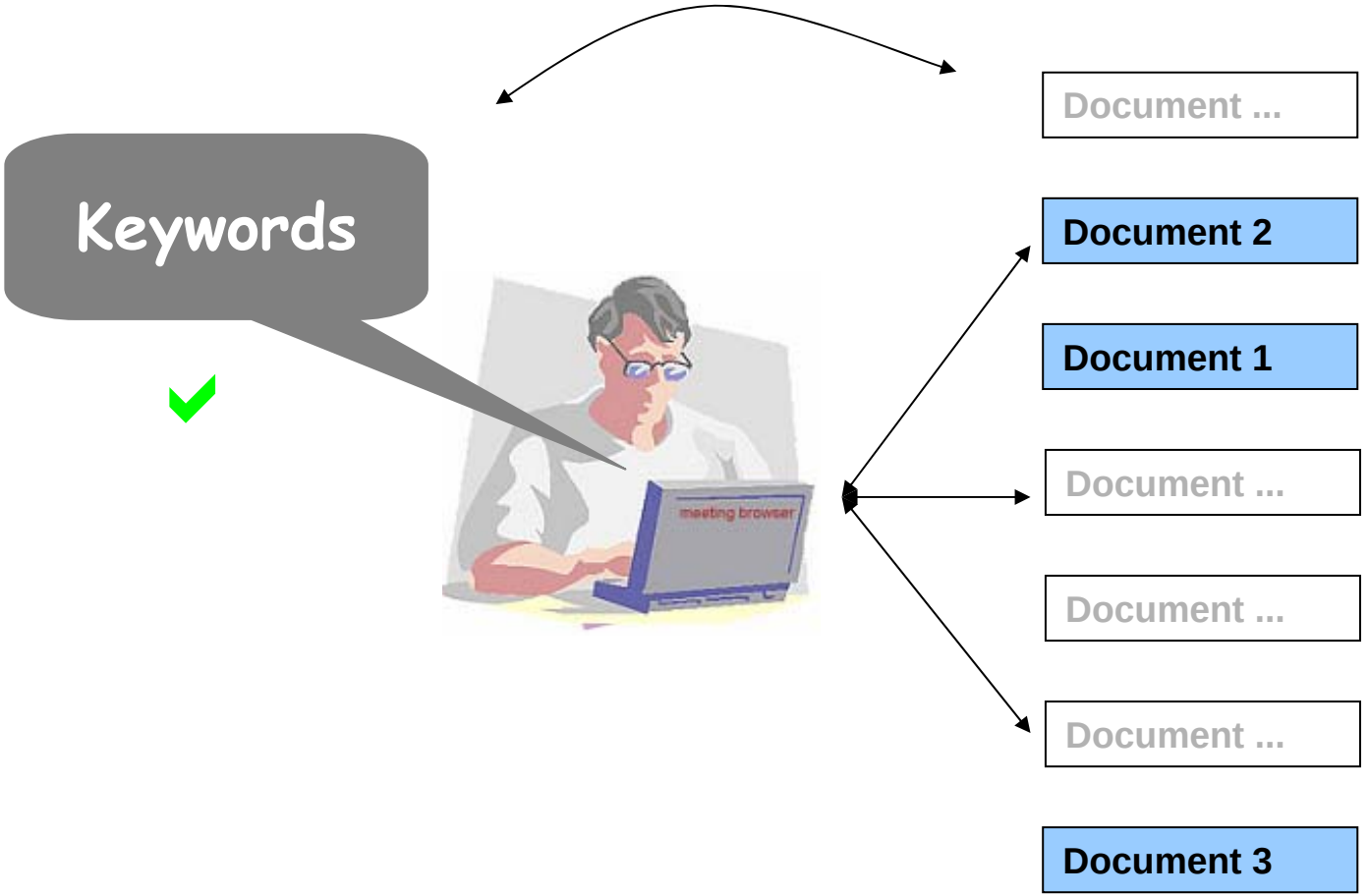


Bridge the human – computer gap



Information Retrieval (IR)

Boolean, Vector Space, ...



SIR-Project

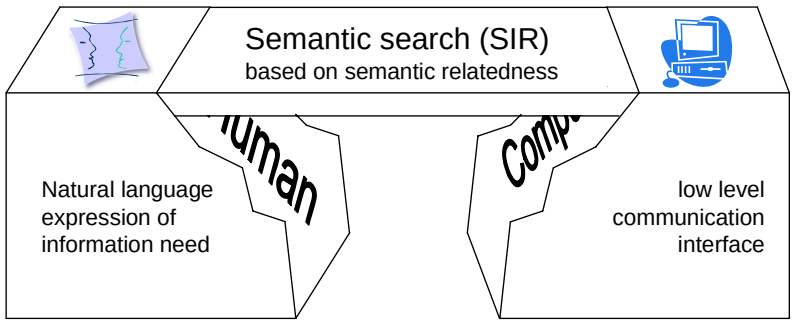
Semantic Relatedness

baker, to program, quality assurance

Essay

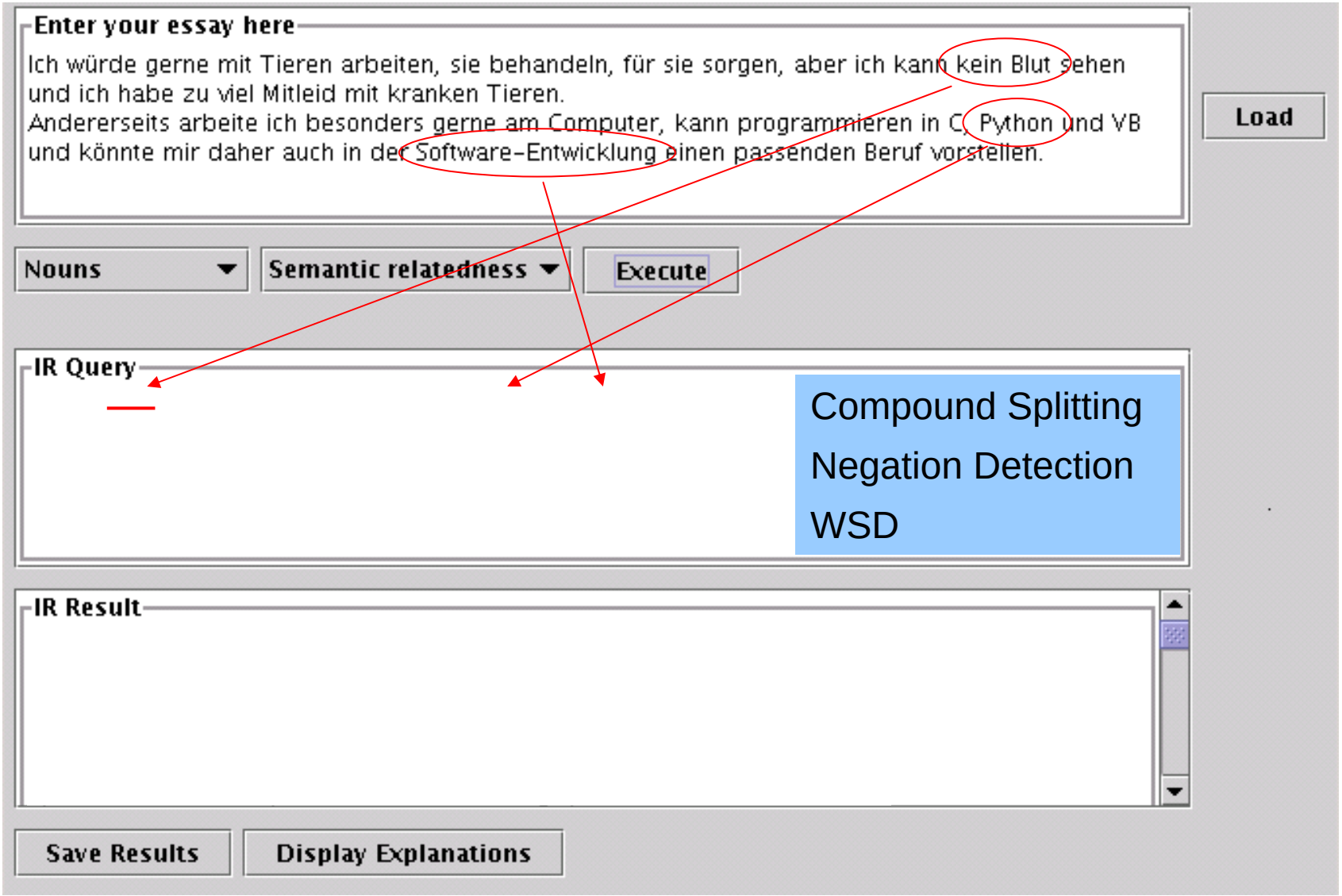
cake, computer, to read, ...

- Profession ...
- **Profession 2**
- **Profession 1**
- Profession ...
- Profession ...
- Profession ...
- **Profession 3**



SIR Example

find good index terms



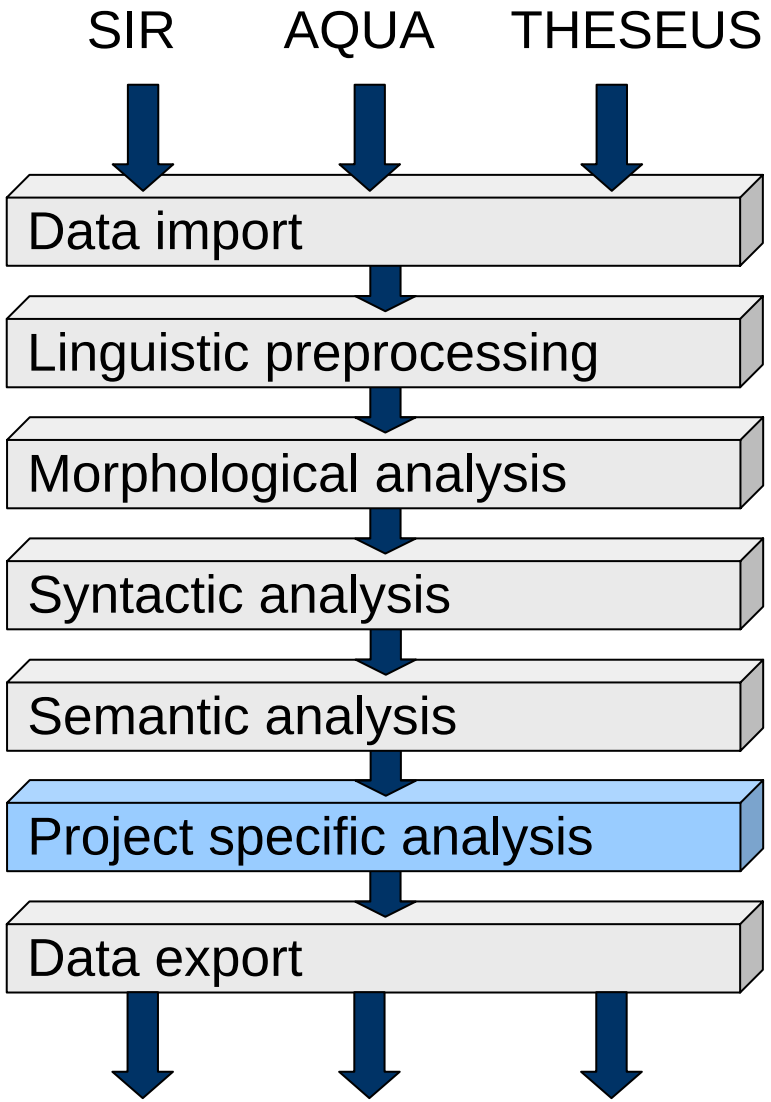
compute semantic relatedness

THESEUS - TEXO

- Large-scale BMBF-Project, industry (SAP, Siemens, etc.)
- Service Marketplaces in Web 2.0
  → Find services, both users and machines
- Problem:
  - Only keyword-based search
  - Lack of ontologies for semantic search
- Solution:
  - Use natural language descriptions of web services
  - Apply Semantic Information Retrieval
  - Community Mining for optimized service selection

  → Darmstadt Knowledge Processing Repository

UIMA components



Wikipedia reader, *Forum reader*, Plain text reader

Tokenizer, Sentence splitter, Stopword tagger

Stemmer, Lemmatizer, Compound Splitter

PoS-Tagger, *Parser*

NE tagger, Sentiment detector, WSD component

Swear word tagger (AQUA), *Negation detection (SIR)*

Indexer (Lucene, Terrier), *ARFF export*

Advantages of UIMA

- Components can be shared between projects
- Shared model of thinking
  - “Reader + Annotators + Consumer”
  - Configuration of components
- Descriptive component orchestration

Challenges

- Agree on a type system
  - No automatic type mapping
- Some rough edges in UIMA
  - No real plug’n’work with PEAR packages
  - Using constraints to align annotations seems to be slow


Wish list

- Automatic type matching
- Better tool support
  - Improving Eclipse plug-ins (robustness, features)
  - Refactoring of UIMA components
  - CPE runner ++ (automatic logging, performance monitor, etc.)
- Plug'n'work approach
- "Import by name" in CPEs
  - Or make ${CPM_HOME}/path also work for readers/consumers
- Construct XML descriptors from Java annotations
- More intuitive API

Thank you very much!

- Acknowledgements:
  - DFG for funding “Semantic Information Retrieval”
  - DFG for funding “Automatic Quality Assessment and Feedback in eLearning 2.0”

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

Deutsche Forschungsgemeinschaft DFG

Ubiquitous Knowledge Processing

TECHNISCHE UNIVERSITÄT DARMSTADT