A Gentle Introduction to UIMA

Björn Rudzewitz

brzdwtz@sfs.uni-tuebingen.de
University of Tübingen

May 13, 2016
Introduction

Conceptual Overview

References
Introduction

- **Unstructured Information Management Architecture**
- **Framework:**
  - define components after pre-defined patterns
  - adapt components for your use
- from unstructured information (text, video, audio, etc.) to structured information (POS sequences, Trees, XML, etc.)
- scalable, highly flexible framework
- IBM Watson
CollectionReader
- reads data
- prepares CAS

* CAS = Common Analysis Structure = View

hierarchical annotation index for a document or a part of a document
what is analyzed together

© 2016 Björn Rudzewitz
UIMA Main class
Pipeline assemble components for UIMA

unstructured information

CollectionReader
- reads data
- prepares CAS*

1 CAS at a time

Analysis Engine (AE)
- analyzes
document
- add annotations**
to Cas Annot. Index
- (potentially)
reuses info from
other AEs

unstructured information

* CAS = Common Analysis Structure = View

hierarchical annotation index for a
document or a part of a document
"what is analyzed together"

© 2016 Björn Rudzewitz
* CAS = Common Analysis Structure = View

Hierarchical annotation index for a document or a part of a document "what is analyzed together"

© 2016 Björn Rudzewitz
**Type System**

explicit hierarchical declaration of all annotation types shared by all modules

- begin
- end
- name
- features

© 2016 Björn Rudzewitz
**CAS** = Common Analysis Structure = View

- InitialView
  - SA
  - TA
  - Q

**Type System**
explicit hierarchical declaration of all annotation types shared by all modules

- Type
  - begin
  - end
  - name
  - features

© 2016 Björn Rudzewitz
1. **CAS = Common Analysis Structure = View**
   - CAS = hierarchical annotation index for a document or a part of a document
   - “what is analyzed together”

2. **Type System**
   - explicit hierarchical declaration of all annotation types shared by all modules
   - Type
     - begin
     - end
     - name
     - features

© 2016 Björn Rudzewitz
UIMA
Main class
Pipeline
assemble components for UIMA

unstructured information

CollectionReader
- reads data
- prepares CAS*

Analysis Engine (AE)
- analyzes document
- add annotations**
to Cas Annot. Index
- (potentially)
reuses info from other AEs

AE
poss. aggregate AEs

Output Consumer
- produces results from gained information / annotations
- FeatureExtraction, MI, Visualization

unstructured information

Sent. - Tok.
POS
NE
- Tok

Sent. - Tok.
POS
NE
- Tok

Sent. - Tok.
POS
NE
- Tok

* CAS = Common Analysis Structure = View
** Type System
explicit hierarchical
declaration of all annotation types
shared by all modules

InitialView
SA
TA
Q

CAS
- AnnotationIndex
- DocumentText
- DocumentLanguage
- Meta Data

CAS

CAS

CAS

CAS

Type
- begin
- end
- name
- features

UIMA : XML Descriptors
+ Java
- implement interfaces of proto modules
- framework calls correct methods automatically

© 2016 Björn Rudzewitz
UIMA
Main class
Pipeline
assemble components for UIMA

unstructured information

JCasCollectionReader_ImplBase
CollectionReader
- reads data
- prepares CAS *

1 CAS at a time

JCasAnnotator_ImplBase
Analysis Engine (AE)
- analyzes document
- adds annotations **
to Cas Annot. Index
- (potentially) reuses info from other AEs

poss. aggregate AEs

JCasAnnotator_ImplBase
Output Consumer
- produces results from gained information / annotations
- FeatureExtraction, ML, Visualization

unstructured information

Sent. - Tok. POS NE - Tok

Sent. - Tok. POS NE - Tok

Sent. - Tok. POS NE - Tok

* CAS = Common Analysis Structure = View

- hierarchical annotation index for a document or a part of a document
- "what is analyzed together"

InitialView

SA TA Q

CAS
- AnnotationIndex
- DocumentText
- DocumentLanguage
- Meta Data

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

CAS

Type
- begin
- end
- name
- features

** Type System

- explicit hierarchical declaration of all annotation types
- shared by all modules

UIMAFit

- implement interfaces of proto modules
- framework calls correct methods automatically

© 2016 Björn Rudzewitz
References

Official UIMA Documentation:
https://uima.apache.org/d/uimaj-current/index.html
(last accessed 12/05/2016)

David Ferrucci and Adam Lally. UIMA: an architectural approach
to unstructured information processing in the corporate research
environment. *Natural Language Engineering*, 10(3-4):327–348,
2004.

Philip Ogren and Steven Bethard. Building test suites for UIMA
components. In *Proceedings of the Workshop on Software
Engineering, Testing, and Quality Assurance for Natural
Language Processing (SETQA-NLP 2009)*, pages 1–4, Boulder,