Computational Argumentation — Orga 1

Organizational Course Information

Henning Wachsmuth

henningw@upb.de

PADERBORN UNIVERSITY

General information

- **Course L.079.05811**

  - Lectures. Henning Wachsmuth

  - Tutorials. Maximilian Spliethöver

  - Languages. English, Python

- **Information**

  - https://cs.upb.de/css/teaching/courses/computational-argumentation-s22

  - https://paul.upb.de and https://panda.upb.de
    → L.079.05811 Computational Argumentation

- **Time and location**

  - Lectures (as of April 12). Tuesday 11–14 c.t., O2

  - Tutorials (as of April 21). Thursday 11–13 c.t., O2

- **Consultation?**

  - Set up appointment with me via e-mail (henningw@upb.de)

Web resources

- **Course web page**
  - General. Detailed course information, general announcements
  - Lectures. Slides

- **PAUL**
  - General. Standard course information
  - Registration. Module, course, course achievement, exam

- **PANDA**
  - General. All announcements, asynchronous Q&A (forum)
  - Lectures and tutorials. Slides, recordings
  - Assignments. Sheets, group submissions, results

- **Video portal (via PANDA)**
  - Recordings. Old recordings from previous years provided
    Note that they may partly be outdated!

Topic

- **This course**

  - Computational analysis and synthesis of natural language arguments
    Introductory overview of the topic today

  - Builds upon natural language processing (NLP)

  - Knowledge of basics in NLP (or at least machine learning) expected
    There will be a high-level recap in one lecture part, but not more

  - Programming skills expected (Python recommended)

- **Recommended courses before (alternatively)**

  - Introduction to Text Mining. Bachelor, Wachsmuth

  - Statistical Natural Language Processing. Master, Ngonga Ngomo

  - Machine Learning 1. Master, Hüllermeier/Ramaswamy

- **Goal of this course**

  - Understand main concepts and methods of an advanced NLP topic

  - Learn to develop computational argumentation algorithms and applications

  - Maybe learn to argue better ;)

Course elements

- **Lectures**
  - Presentation of course content (and organizational info)

- **Tutorials**
  - Presentation of assignments and solutions, Q&A on assignments and other

- **Assignments (also see next slide)**
  - Sophisticated programming tasks related to lecture topics
  - Amount. 4 in total, 2 weeks each (1-week breaks in-between)
  - Group work. You need to submit in groups of 2–4 people
  - Submission. You need to submit code, documentation, and instructions

  - Evaluation. Graded as convincing (A), working (B), or not working (F)
    See next slide on course achievement!
  - Bonus. (a) Min. 3x B: exam grade + 1/3, (b) 4x A/B, min. 2x A: grade + 2/3
    For example, in case of (b) an exam grade of 2.7 is changed to 2.0; only grades better than 5.0 can be improved.

How to complete the course

- **Examination**
  - Course achievement. Need at least 2x B in programming assignments
  - Oral exam. Mainly on content of the lectures
    Details on exam later. Tentative dates: (1) end of July; (2) second week of September or later

- **Four registrations needed**
  - Module + course. Both until April 22
  - Course achievement. April 18 – May 18 (de-registration until July 1)
  - Examination. April 18 – May 18 (1st phase), August 29 – September 2 (2nd)
    Cancellation until one week before examination takes place

- **How to register**
  - All registrations are done in PAUL, requiring two clicks (”Register“, ”Submit“)
   General rule: If you see anything in PAUL that you can register for, do so
  - Regularly check emails to your PAUL email address
  - If you need advice, contact study-cs@mail.upb.de or see office hours:
    https://cs.upb.de/studium/beratung-und-unterstuetzung/fachberatung/

Literature and code basis

- **Books (not obligatory)**

    - General NLP books
      (Jurafsky and Martin, 2009; Wachsmuth, 2015)

    - Argumentation Mining
      (Stede and Schneider, 2018)

    - Few exemplars in library
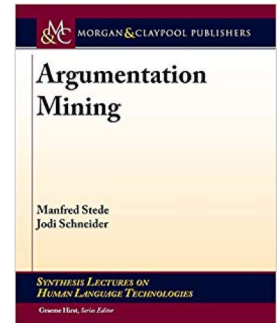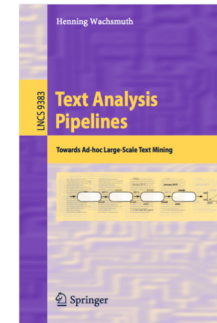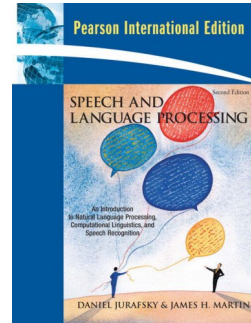
- **Conference and journal papers**

    - References to papers will occur in course content

    - Most papers can be found online (e.g., at https://www.aclweb.org/anthology/)

- **Code**

    - Different general NLP libraries available freely
      github.com/stanfordnlp/stanza/, www.nltk.org, spacy.io, pypi.org/project/polyglot/, github.com/zalandoresearch/flair

    - Papers often provide a URL where code can be found

    - Still, extensive own implementation needed in programming tasks

Lecture schedule

- **Basics**
  - Apr 12   Introduction to computational argumentation
  - Apr 19   Basics of natural language processing
  - Apr 26   Basics of argumentation
  - May 3   Argument acquisition

- **Methods**
  - May 10–17   Argument mining
  - May 24–31   Argument assessment
  - Jun 7–14   Argument generation

- **Applications**
  - Jun 21–28   Applications of computational argumentation
  - Jul 5   Conclusion

- **Notice**
  - No lecture on July 12

Assignment and tutorial schedule

- **Introduction (tutorial concept, python recap)** on April 21

- **Assignment 1: Argument acquisition**
  - Duration. April 25 (publication) – May 9 (submission)
  - Tutorials. April 28 (task), May 5 (Q&A), May 19 (solution)

- **Assignment 2: Argument mining**
  - Duration. May 16 (publication) – May 30 (submission)
  - Tutorials. May 19 (task + Q&A), June 9 (solution)   (holiday on May 26)

- **Assignment 3: Argument assessment**
  - Duration. June 6 (publication) – June 20 (submission)
  - Tutorials. June 9 (task + Q&A), June 30 (solution)   (holiday on June 16)

- **Assignment 4: Argument generation**
  - Duration. June 27 (publication) – July 11 (submission)
  - Tutorials. June 30 (task), July 7 (Q&A), July 14 (solution)

- **Conclusion (exam questions)** on July 14

References

- **Jurafsky and Martin (2009).** Daniel Jurafsky and James H. Martin (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition, Prentice-Hall, 2009.
  (free draft of 3rd edition here: https://web.stanford.edu/~jurafsky/slp3/)

- **Stede and Schneider (2018).** Manfred Stede and Jodi Schneider. Argumentation Mining. Synthesis Lectures on Human Language Technologies 40, Morgan & Claypool, 2018.

- **Wachsmuth (2015).** Henning Wachsmuth. Text Analysis Pipelines — Towards Ad-hoc Large-scale Text Mining. LNCS 9383, Springer, 2015.
  (free preprint here: http://argumentation.bplaced.net/arguana-publications/papers/wachsmuth15a-springer-preprint.pdf)