Computational Argumentation

Organizational Course Information

Henning Wachsmuth
henningw@upb.de

April 10, 2019

PADERBORN
UNIVERSITY

Course: Meta

- **Meta**
  - Course number. L.079.05811
  - Instructors. Henning Wachsmuth (and Milad Alshomary)
  - Languages. English, Python

- **Course material**
  - https://cs.upb.de/css/teaching/courses/computational-argumentation-s19/
  - http://paul.upb.de

- **Dates and locations**
  - Lectures. Wednesday 13–16 c.t. (as of April 10), in O1.258
  - Tutorials. Tuesday 9–11 c.t. (as of April 16), in F1.520

  Notice: Tutorial next week Tuesday at 9 s.t. in H5 (campus)

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

Course: Topic

- **This course: Computational Argumentation**
  - Computational analysis and synthesis of natural language argumentation
    Introduction to the topic today
  - Builds upon natural language processing (NLP)
  - Knowledge of basics in NLP (or at least machine learning) expected
    There will be a recap of main concepts and methods in one lecture, but not more
  - Programming skills expected (Python recommended)

- **Recommended courses to take before (alternatively)**
  - Introduction to Text Mining. Bachelor, Wachsmuth
  - Statistical Natural Language Processing. Master, Ngonga Ngomo
  - Machine Learning 1. Master, Hüllermeier

- **Goal of this course**
  - Understand main concepts and methods of an advanced NLP topic
  - Learn to develop computational argumentation algorithms and applications
  - Learn to teach... and to better argue?

Course: Concept

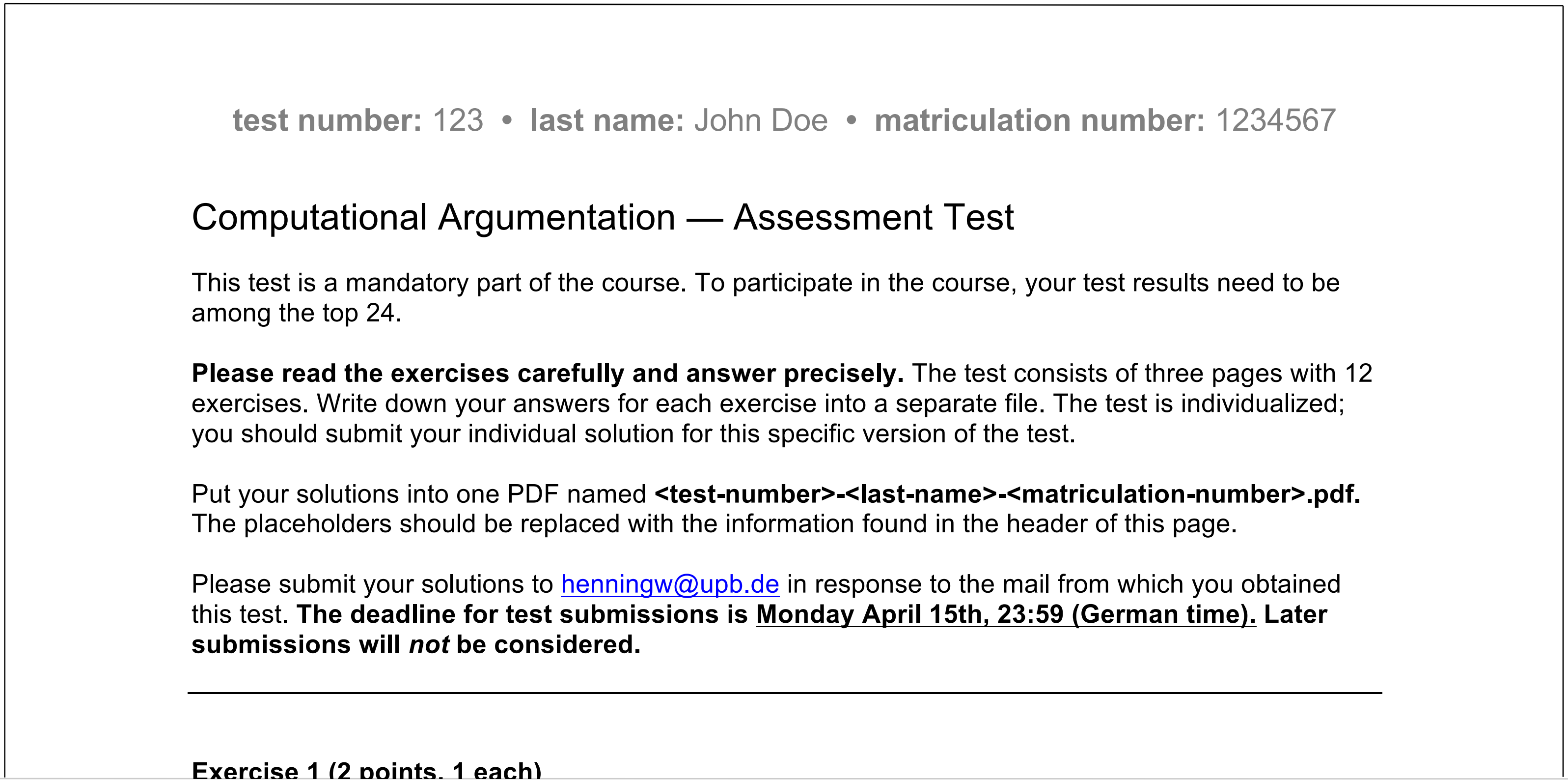
- **Schedule (details below)**
  - First half. Regular lectures and tutorials held by me
  - Second half. Lectures and tutorials held by students (you!)
- **Examination (details below)**
  - No oral or written exam!
  - Instead: Grading of lecture (~60%) and programming tutorial (~40%)

    Both lecture and tutorial need to be passed (4.0 or better) individually!
- **Group work**
  - You need to work in groups of 3 (in exceptional cases, in groups of 2)
  - You can indicate who you like to work with

    Groups assigned by me based on this and based on topic preferences (as far as possible)
- **Limited number of participants (details below)**
  - By concept, number of possible participants limited (max. 24)
  - Consequence: Initial assessment test to decide about participation

Assessment test

- **Everyone who is registered today until 5 pm**
  - Will get an e-mail with an individualized test to the address specified in PAUL
  - Should reply to the e-mail **by Monday April 15, 23:59** with the solutions
  - Participation will be decided then, topic assignment follows next



test number: 123 • **last name:** John Doe • **matriculation number:** 1234567

Computational Argumentation — Assessment Test

This test is a mandatory part of the course. To participate in the course, your test results need to be among the top 24.

Please read the exercises carefully and answer precisely. The test consists of three pages with 12 exercises. Write down your answers for each exercise into a separate file. The test is individualized; you should submit your individual solution for this specific version of the test.

Put your solutions into one PDF named **<test-number>-<last-name>-<matriculation-number>.pdf.** The placeholders should be replaced with the information found in the header of this page.

Please submit your solutions to henningw@upb.de in response to the mail from which you obtained this test. **The deadline for test submissions is Monday April 15th, 23:59 (German time). Later submissions will *not* be considered.**

Exercise 1 (2 points, 1 each)

Course: Tasks and grading

- **Required course achievement** (”Studienleistung“)
  - Being among the top 24 that pass the initial assessment test
    Deadline is next Monday
  - Presence and activity in > 50% of all lectures and tutorials held by students
- **Graded tasks (for groups of 3)**
  - Prepare and give 2.5 hours lecture, incl. 15 minutes break
    incl. lecture slides as PDF (and permission to share)
  - Prepare and give 1.5 hours programming tutorial
    incl. programming task basis as Python source code (and permission to share)
  - Meet deadlines specified below
    All deadlines mean ”until 23:59 German time“ on the respective day
- **Notice: This is NOT a seminar**
  - Lectures should enable other students to work on the respective topic
    Details on scientific presentation given next Tuesday
  - Tutorials need to provide code environment that allows practicing the topic
    Example tutorial given by us before

Registration for modules, courses, and examinations

- **To complete this course, you need to register 4 times**
  - Until 5 pm today (April 10). For the module *and* for the course
  - From April 15 to May 15. For the course achievement *and* for the examination

  - All registrations are done in PAUL, requiring two clicks (”Register“, ”Submit“).
  - General Rule. If you see anything in PAUL that you can register for within this course or module, you should do so.

- **To prevent things from going wrong**
  - All information necessary is available in PAUL — somewhere.
  - Regularly check the email address that PAUL sends its messages to.
  - If anything looks suspicious in PAUL, contact the examination office.
  - If you need advice, contact study-cs@mail.uni-paderborn.de.
  - Visit https://cs.upb.de/studium/beratung-und-unterstuetzung/fachberatung/ for contact and office hours of the advisory service.

Notice: This information comes from the student advisory service

Course: Schedule (1 of 2)

- **Teacher giving lectures and tutorials**
  - Wed, Apr 10 Lecture. Introduction to computational argumentation
  - Mon, Apr 15 **Deadline.** Submission of assessment tests (result until next lecture)
  - Tue, Apr 16 Tutorial. Introduction to scientific presentation

  - Wed, Apr 17 Lecture. Basics of natural language processing
  - Mon, Apr 22 **Deadline.** Submission of topic preferences (result until next lecture)
  - Tue, Apr 23 Tutorial. Basics of natural language processing

  - Wed, Apr 24 Lecture. Basics of argumentation
  - Tue, Apr 30 Tutorial. Basics of argumentation

  - Wed, May 1 No lecture (holiday)
  - Tue, May 7 Tutorial. Applications of computational argumentation

  - Wed, May 8 Lecture. Resources for computational argumentation
  - Tue, May 14 Tutorial. Resources for computational argumentation

Course: Schedule (2 of 2)

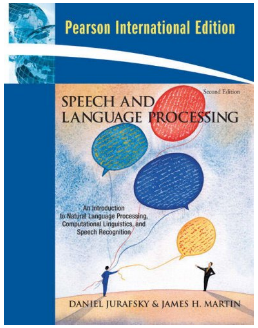
- **Students giving lectures and tutorials (as of May 15)**
  - Wednesdays Lecture. On one particular topic
  - Tuesdays Tutorial. Programming task on the same topic
  - 7 days before **Deadline.** Submission of lecture+tutorial draft (feedback will be given)

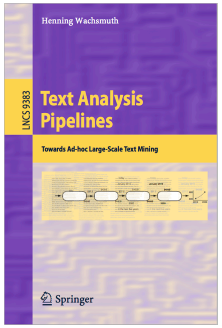
1.	May 15+21	Mining of argumentative units
2.	May 22+28	Mining of supporting and objecting units
3.	May 29+Jun 4	Mining of argumentative structure
4.	Jun 5+11	Assessment of the structure of argumentation
5.	Jun 12+18	Assessment of the reasoning of argumentation
6.	Jun 19+25	Assessment of the quality of argumentation
7.	Jun 26+Jul 2	Generation of argumentation
8.	Jul 3+9	Development of an argument search engine

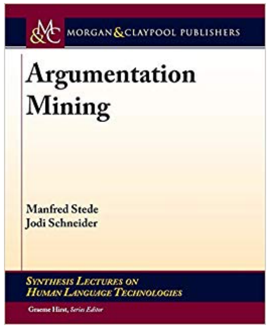
- **Teacher giving lecture**
  - Jul 10 Conclusion

Literature and code basis

- **Books**
  - General NLP books
    (Jurafsky and Martin, 2009; Wachsmuth, 2015)
  - ”Argumentation Mining“
    (Stede and Schneider, 2018)
  - Few exemplars in library
  - One exemplar of each can be issued from me briefly for copying purposes


Pearson International Edition
SPEECH AND LANGUAGE PROCESSING
Second Edition
An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition
DANIEL JURAFSKY & JAMES H. MARTIN




M&C MORGAN & CLAYPOOL PUBLISHERS
Argumentation Mining
Manfred Stede
Jodi Schneider
Synthesis Lectures on Human Language Technologies
Graeme Hirst, Series Editor

- **Conference and journal papers**
  - Some paper references to start with will be given next week
  - Most papers can be found online (e.g., at aclanthology.info)

- **Code**
  - Different general NLP libraries available freely
    stanfordnlp.github.io/stanfordnlp/, 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 likely to be needed

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.
- **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.