Hertie School Fall Semester 2025

Course Syllabus

GRAD-E1282: Natural Language Processing

1. General information

Course Format	Lecture, Online, Wednesday 12pm-2pm, via Teams Lab, On-site Lab 1: Thursdays, 12pm-2pm, R2.34 Lab 2: Fridays, 10am-12pm, R2.34
Instructors	Dr. Sascha Göbel, Luis Fernando Ramírez Ruiz
Instructor's E-mail	goebel@hertie-school.org
Instructor's Office Hours	Via Teams by appointment.

Link to Study, Examination and Admission Rules and MIA, MDS and MPP Module Handbooks

For information on **course room, times and session dates,** please consult the Course Plan on *MyStudies*.

Instructor Information:

Sascha Göbel is a postdoctoral researcher at the Hertie School Data Science Lab. Before joining the Data Science Lab, he worked as a postdoctoral researcher at Goethe University Frankfurt and received his doctoral degree in Political Science from the University of Konstanz. His research focuses on digital information and communication technologies in politics, political behavior, and public opinion, with an emphasis on the application and development of statistical and data-intensive computational approaches.

2. Course Contents and Learning Objectives

Course contents:

This course provides a foundational and practical introduction to natural language processing (NLP). Students will engage with essential NLP concepts and algorithmic tools, understand how and why to preprocess unstructured textual data, and learn about different text representations. On the applied side, the course communicates the motivation and inner workings of core NLP models, including both basic as well as more complex neural models, and how to program, implement, and evaluate such models using Python. Students practice applying this knowledge in the context of their own research questions as well as in an accompanying lab. This is an advanced course intended for students with prior knowledge of statistics at the level of "Statistics II: Statistical Modeling and Causal Inference" and prior knowledge of working with R or Python at the level of "Introduction to Data Science".

Main learning objectives:

At the end of this course, you should:

(1) Have acquired a fundamental understanding of the logic, functionality, and rationale behind core NLP approaches.
(2) Have a better grasp of essential NLP concepts, algorithmic tools, and assumptions and how to appropriately build NLP systems.
(3) Be able to specify and implement fundamental NLP applications using Python.
(4) Know how to interpret, evaluate, and effectively communicate NLP applications.
(5) Be able to explore and develop more complicated/realistic NLP systems independently or in the context of further seminars.

Software:

To work through this course, a working installation Python is required. In class, we will work with the integrated development environment PyCharm. All software is available online for free. Detailed instructions for downloading and installing the required software will be made available via Moodle.

Target group:

Students should have a keen interest in statistical inference in a social scientific context as well as in learning new programming languages.

Teaching style:

This is a methods course that combines lecture-style teaching with the practical and joint implementation of code in Python during separate lab session. The lecture is taught by Sascha Göbel. The lab sessions are taught by Luis Fernando Ramírez Ruiz. Students are encouraged to ask questions at any time and are also confronted with questions by the lecturer during the course.

Prerequisites:

We assume prior knowledge of statistics at the level of "Statistics II: Statistical Modeling and Causal Inference", including basic probability principles, familiarity with standard notation for statistical inference, (generalized) linear regression, and basic matrix operations. As regards programming, prior knowledge of the R or Python programming languages at the level of "Introduction to Data Science" is expected. Knowledge of control structures, i.e., if-else and loops, as well as data types and structures is essential.

3. Grading and Assignments

Composition of Final Grade:

Assignment	Due	Submission	Grade
Assignment 1: Class Participation	-	Moodle	10%
Assignment 2: Quizzes (Q)	Deadline: Wednesday in the week after the	Moodle	5 x 9%

	seminar until 11:59pm		
Assignment 3: Research Note	Deadline: 23 December until 11:59pm	Submit via Moodle	45%

Assignment Details:

Assignment 1: Class participation
Content discussed in class assumes a thorough prior reading of the assigned required literature and students are confronted with questions by the lecturer during the course. The participation grade probes this knowledge and assumes that students take part, not as passive consumers, but as active participants, including the exchange, production, and critique of ideas—their own ideas and the ideas of others. Therefore, students should come to class having read the materials assigned for that day and contribute thoughtfully to the conversation. Participation is marked by its active nature, its consistency, and its quality.

Assignment 2: Quizzes
Starting from session 2 and until session 10, every second session, an online quiz, contributing 9% to the final grade, will assess your knowledge and comprehension related to the material of the last few sessions. A "Q" in the session overview below indicates quiz dates.

Assignment 3: Research note
In their research note, students will engage in practical natural language processing using R or Python in the context of one of two scenarios, contributing 45% to the final grade. In scenario 1, students search for and choose a published study that employs textual data. Relying on the available replication data, students extend the analysis using some of the approaches discussed in class. In scenario 2, students pursue their own research question with data of their choosing using the steps and tools covered throughout this course. Students are free to choose one of the two scenarios for this assignment. Findings must be reported in a research note (approx. 10 pages, excluding the title page and references, and no more than 2 figures in main text). It is important to start working on the assignment as early as possible. Students are strongly encouraged to discuss their plans for the research note with the instructor during the office hour.

Late submission of assignments:

For each day the assignment is turned in late, the grade will be reduced by 10% (e.g. submission two days after the deadline would result in 20% grade deduction).

Attendance:

Students are expected to be present and prepared for each class session. Active participation during lectures and seminar discussions is essential. Please note that students can miss up to two sessions (out of twelve) if no course assignments are affected. For further information please consult the Examination Rules §10.

Academic Integrity:

The Hertie School is committed to the standards of good academic and ethical conduct. Any violation of these standards shall be subject to disciplinary action. Plagiarism, misuse of AI, free riding in group work, and other deceitful actions are not tolerated. See Examination Rules §16, the Hertie Plagiarism Policy, and the Hertie Guidelines for Artificial Intelligence Tools.

Compensation for Disadvantages:

If a student furnishes evidence of being unable to take an examination as required in whole or in part due to disability or permanent illness, the Examination Committee may upon written request approve learning accommodation(s). In this respect, the submission of adequate certificates may be required. See Examination Rules §14.

Extenuating circumstances:

An extension can be granted due to extenuating circumstances (i.e., for reasons like illness, personal loss or hardship, or caring duties). In such cases, please contact the course instructor and Examination Office *in advance* of the assignment deadline.

4. Session Overview

Session	Session Title
1	Motivation and course overview
2	Basic string processing and regular expressions (Q)
3	Text preprocessing
4	Text representation I - Vector-based (Q)
5	Text representation II - Distribution-based
6	Basic models I - Text classification with logistic regression (Q)
7	Basic models II - Topic modelling with LDA
8	Neural models I - Text classification with neural networks (Q)
9	Neural models II - Language modelling with Transformers I
10	Neural models II - Language modelling with Transformers II
11	NLP visualizations
12	Extending NLP to LLM workflows and agents

5. Course Sessions and Readings

All mandatory course readings can be accessed on the course Moodle page.

Session 1: Motivation and course overview	
During the first meeting, we will explore the definition of and rationale behind natural language processing and discuss potential applications. Toward the end of the session, we will look ahead and clarify organizational matters.	
Required Reading	- Vajjala et al. (2020). Chapter 1
Optional Reading	- Hobson/Dyshel (2024). Chapter 1, 1-1.3.1

Session 2: Basic string processing and regular expressions	
This session is fully dedicated to regular expressions, the fundamental tool for describing text patterns. Regular expressions offer an algebraic notation for characterizing a set of strings and lie at the heart of text normalization and many algorithmic tools in NLP.	
Required Readings	- Jurafsky/Martin (2024). Section 2.1
Optional Readings	- Hobson/Dyshel (2024). Chapter 1, 1.4-1.6.1
Core Python Tools	re

Session 3: Text preprocessing	
In this session, we discuss the different steps involved in text normalization, the process of converting unstructured text data to a more convenient format. Starting from collections of text, this includes tokenization, the procedure to separate out words from running text, the normalization of word formats, and the segmentation of texts into sentences or other meaningful units.	
Required Readings	- Hobson/Dyshel (2024). Chapter 2, 2-2.9
Optional Readings	- Jurafsky/Martin (2024). Sections 2.2-2.7
Core Python Tools	nltk, sentencepiece, scikit-learn, spacy, tokenizers

Session 4: Text representation I - Vector-based	
In our first session on text representation, we begin to develop an intuition for representing text numerically. To this end, we consider common vectorization approaches, such as One-Hot-Encoding, Bag Of Words, and TF-IDF. Following a conceptual introduction, we implement these text representation approaches using core python libraries for NLP.	
Required Readings	- Vajjala et al. (2020). Chapter 3, “Basic Vectorization Appoaches”
Optional Readings	- Hobson/Dyshel (2024). Chapter 3
Core Python Tools	numpy, pandas, scikit-learn

Session 5: Text representation II - Distribution-based	
We continue our inquiry into text representations and build on the previous session by focusing on distribution-based representations. In particular, we consider low-dimensional learned text representations called embeddings, their architectural variants, and how to implement them in Python.	
Required Readings	- Vajjala et al. (2020). Chapter 3, "Distributed Representations" and "Distributed Representations Beyond Words and Characters"
Optional Readings	- Hobson/Dyshel (2024). Chapter 6
Core Python Tools	fasttext, gensim

Session 6: Basic models I - Text classification with logistic regression	
This session introduces basic yet highly useful and widely applied supervised models for natural language processing. We discuss different classification approaches, such a logistic regression and support vector machines, and demonstrate the practical setup and training routine in the context of logistic regression applied to text classification tasks.	
Required Readings	- Vajjala et al. (2020). Chapter 4, up to and including "Support Vector Machine"
Optional Readings	- Raschka et al. (2022). Chapter 3, "Modeling class probabilities via logistic regression"
Core Python Tools	imblearn, scikit-learn

Session 7: Basic models II - Topic modelling with LDA	
The second session on basic models for NLP introduces unsupervised models. Following a brief conceptual introduction of different algorithms, we focus on Latent Dirichlet allocation and its practical implementation to discover topics or meaning in texts.	
Required Readings	- Hobson/Dyshel (2024). Chapter 4.
Optional Readings	- Raschka et al. (2022). Chapter 8, "Topic modelling with latent Dirichlet allocation"
Core Python Tools	numpy, pandas, scikit-learn

Session 8: Neural models I - Text classification with neural networks	
In this session we begin delving into more complex neural models. We start with a brief conceptual introduction of neural networks in general and 1-dimensional convolutional neural networks for text in particular. Finally, we introduce PyTorch by implementing the latter for the purpose of text classification.	
Required Readings	- Raschka et al. (2022). Chapter 12, "First Steps with PyTorch", "Building input pipelines in PyTorch", and "Building an NN model in Pytorch"; Chapter 13, "Simplifying implementations of common

	architectures via the torch.nn module"; Chapter 14, "The building blocks of CNNs"
Optional Readings	- Raschka et al. (2022). Chapter 2 and 11
Core Python Tools	torch

Session 9: Neural models II - Language modelling with Transformers I	
Further increasing the complexity of neural models for NLP, we begin discussing the anatomy of transformers with a particular focus on their precursors - recurrent neural networks for sequence modelling. On the practical side, we extend our PyTorch-based implementation of a neural network model for text classification from the previous section by adding corresponding RNN and LSTM layers.	
Required Readings	- Raschka et al. (2022). Chapter 15 - Turnstall et al. (2022). Chapter 1, "The Encoder-Decoder Framework", "Attention Mechanisms"
Optional Readings	- Tunstall et al. (2022). Chapter 3
Core Python Tools	torch

Session 10: Neural models II - Language modelling with Transformers II	
In our final session on neural models, we delve deeper into transformers and introduce the Hugging Face ecosystem. Using the corresponding transformers library, we learn how to easily apply and fine tune, i.e., apply transfer learning to, pre-trained large language models for the purpose of text classification.	
Required Readings	- Tunstall et al. (2022). Chapter 1, "Transfer Learning in NLP", "Hugging Face Transformers", and "The Hugging Face Ecosystem"
Optional Readings	- Tunstall et al. (2022). Chapter 2
Core Python Tools	torch, transformers

Session 11: NLP visualizations	
In this session, we turn to visualization techniques for natural language processing. These range from confusion matrices, over word clouds and 2-dimensional embedding visualizations, to local interpretable model-agnostic explanations that allow for exploring the contribution of different textual components to a classification decision.	
Core Python Tools	captum, matplotlib, scikit-learn, tensorboard, wordcloud

Session 12: Extending NLP to LLM workflows and agents	
The final session explores how modern NLP techniques can be extended into broader AI applications by orchestrating large language models in workflows and agents. Core concepts of LangChain and LangGraph are introduced and illustrated through minimal, hands-on examples.	
Required Readings	- TBA

Optional Readings	- TBA
Core Python Tools	langchain, langgraph

Final Exam Week: no class