Subject name/code:	**Introduction to language technology / BNY-DK-155A**
Subject coordinator:	Head of the Doctoral School
Lecturer(s) of the subject:	Noémi Ligeti-Nagy, PhD
Credits:	8 credits
Lesson type:	lecture
Brief subject description:	The course provides a practical overview of computational methods used in empirical linguistic analysis and natural language processing. It introduces central concepts and techniques in computational linguistics, language technology, and NLP, including regular expressions, corpus creation, linguistic annotation, benchmarking, computational semantics, machine translation, neural language models, large language models, and prompt programming. The course combines theoretical discussion with hands-on work. Students read selected chapters and papers, complete individual weekly assignments, and work on a group project in which they build a small annotated corpus and use it for experimentation with language-model fine-tuning. By the end of the course, students will understand the main methodological principles of computational linguistics and will be able to design, document, annotate, evaluate, and present a small-scale language technology project.
Theoretical knowledge to be acquired:	Students will acquire theoretical knowledge of the main areas of computational linguistics and natural language processing, including: • the relationship between linguistics, computational linguistics, language technology, and NLP; • regular expressions and their role in text processing; • corpora, corpus design, data collection, and benchmarking; • annotation theory, annotation guidelines, and inter-annotator agreement; • basic concepts of computational semantics, including word meaning, semantic similarity, word sense disambiguation, and semantic resources; • supervised and unsupervised learning in linguistic analysis; • machine translation and neural approaches to language processing; • language modelling, pretraining, fine-tuning, and large language models; • prompt engineering and the practical use of generative AI tools; • methodological principles of evaluation in NLP and empirical language technology.

Practical knowledge to be acquired:	Students will acquire practical skills in: • using regular expressions for text processing tasks; • collecting, organizing, and documenting linguistic data; • designing annotation categories and annotation guidelines; • carrying out pilot annotation and improving guidelines based on the results; • evaluating annotation consistency and dataset quality; • preparing a small corpus for a computational linguistics task; • interpreting benchmark datasets and evaluation methods; • experimenting with language-model fine-tuning or related language technology workflows; • using selected NLP tools and online resources; • writing a short research paper presenting the literature, methodology, data, and results of a language technology project; • presenting project progress and final results orally.	
List of the most important required literature (2–4 pieces) with bibliographical details (author, title, edition or specific pages, ISBN)	Jurafsky, Daniel, and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 3rd ed. draft. Selected chapters: Chapter 2, Sections 2.1–2.2; Chapter 3, Sections 3.1–3.2.1; Chapter 4, Introduction; Chapter 6, pp. 1–10. Available online. Jurafsky, Daniel, and James H. Martin. *Speech and Language Processing*. Earlier edition, 2014 version. Chapter 1. Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding." *Proceedings of the 2018 EMNLP Workshop BlackboxNLP*, 353–355. Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems." *Advances in Neural Information Processing Systems* 32.	
List of the most important recommended literature (2–4 pieces) with bibliographical details (author, title, edition or specific pages, ISBN)	Wissler, Lars, Mohammed Almashraee, Dagmar Monett, and Adrian Paschke. 2014. "The Gold Standard in Corpus Annotation." *Proceedings of the 5th IEEE Germany Student Conference*. Yeomans, Michael, Alejandro Kantor, and Dustin Tingley. 2018. "The Politeness Package: Detecting Politeness in Natural Language." *The R Journal* 10(2): 489–502. Arafa-Hilal, Marwa. 2023. "The Use of Politeness Strategies in Academic Conversations as Represented in a Corpus Linguistics MOOC." *Journal of Pragmatic Research* 5(1): 85–106. DeepLearning.AI. *ChatGPT Prompt Engineering for Developers*. Short course. Recommended for the class on prompt programming and applied work with ChatGPT.	
Theory to practice ratio: 10 % practice – 90 % theory	Theory lessons: 100%	Practice lessons: 0%

Applied teaching methods:	The course is taught as a seminar combining short lectures, guided discussion, reading-based preparation, practical exercises, and project-based learning. Teaching methods include: • introductory lectures on core concepts and methods; • discussion of required readings and selected research papers; • in-class practical exercises, including regular-expression tasks and small-scale NLP activities; • weekly individual assignments; • group work on corpus creation, annotation, and experimentation; • pilot annotation and discussion of annotation guidelines; • student presentations on benchmark datasets and project progress; • guest lecture on machine translation; • guided exploration of ChatGPT and prompt engineering; • final oral presentation of project results. Students are expected to bring a notebook/laptop to class, as some sessions include practical computational work.
Method of assessment:	Assessment is based on continuous coursework and a final group project. Students complete regular individual assignments during the semester, approximately one assignment per week. The final project consists of building a small corpus for a selected computational linguistics task, documenting the annotation process, evaluating the dataset and/or model results, and presenting the work. The final submission is a 4–6-page paper summarizing the relevant literature, the project methodology, the data and annotation process, and the results obtained. Students also present their project progress during the semester and give a final presentation at the end of the course.
Assessment criteria:	- 80–100% = Excellent (5) - 60–79% = Satisfactory (3) - 0–59% = Unsatisfactory (1)
How the subject contributes to the achievement of the learning outcomes at level 8 of the HQF (MKKR), as identified as learning outcomes in the doctoral school's training programme:	**Knowledge:** The subject contributes to doctoral-level knowledge by introducing students to the theoretical foundations and current methods of computational linguistics and empirical language technology. Students learn how linguistic questions can be operationalized as computational tasks, how corpora and benchmarks are constructed, and how language models and NLP systems are evaluated. The course also familiarizes students with recent developments in neural language modelling, fine-tuning, and large language models, enabling them to situate their own research in relation to contemporary computational approaches. **Skills:** The course develops students' ability to design and implement a small-scale empirical language technology project. Students learn to collect and structure linguistic data, formulate annotation categories, write annotation guidelines, conduct pilot annotation, assess annotation quality, interpret benchmark datasets, and present methodological decisions in academic form. The final paper and presentation strengthen students' ability to communicate research design, data, methods, and results clearly to a scholarly audience.

	Attitudes: The subject promotes a critical and reflective attitude toward computational methods in linguistic research. Students are encouraged to evaluate the reliability, limitations, and biases of corpora, annotations, benchmarks, language models, and generative AI systems. The course also supports openness to interdisciplinary methods, combining linguistic expertise with computational thinking, empirical evaluation, and collaborative research practice. **Responsibility and autonomy:** The course supports doctoral-level autonomy by requiring students to make independent decisions about data selection, annotation design, methodological choices, and interpretation of results within a supervised project framework. Students take responsibility for the quality, documentation, and reproducibility of their work, while also collaborating with peers on a shared corpus-building and annotation task. Through weekly progress reports and the final paper, students practice accountable, transparent, and research-oriented project work.