

## Advanced Language Processing with Python (A704065)

Due to Covid 19, the education and evaluation methods may vary from the information displayed in the schedules and course details. Any changes will be communicated on Ufora.

<b>Course size</b>	<i>(nominal values; actual values may depend on programme)</i>			
<b>Credits</b> 5.0	<b>Study time</b> 150 h	<b>Contact hrs</b>	45.0 h	
<b>Course offerings and teaching methods in academic year 2021-2022</b>				
A (semester 2)	English	Gent	self-reliant study activities	22.5 h
<b>Lecturers in academic year 2021-2022</b>				
Tezcan, Arda		LW22	lecturer-in-charge	
<b>Offered in the following programmes in 2021-2022</b>			<b>crdts</b>	<b>offering</b>
<a href="#">Master of Arts in Technology for Translation and Interpreting</a>			5	A
<a href="#">Master of Arts in Advanced Studies in Linguistics (main subject Natural Language Processing: Theory and Practice)</a>			6	A
<a href="#">Postgraduate Certificate Computer-Assisted Language Mediation</a>			5	A

### Teaching languages

English

### Keywords

Programming, Python, natural language processing, machine learning

### Position of the course

Advanced Language Processing with Python builds on previously acquired programming competencies in Python and focuses on the applications of the Natural Language Processing (NLP) techniques, which are thought in the course "Natural Language Processing". In this course, the students will learn to integrate NLP libraries (such as the Natural Language Toolkit (NLTK), spaCy, NumPy and Scikit-Learn) into Python code and tackle NLP problems using supervised machine learning techniques, such as linear and logistic regression.

### Contents

The course deals with the following topics:

- searching and manipulating text using regular expressions
- integrating NLP libraries into Python
- text pre-processing: tokenization, part-of-speech tagging, named entity recognition
- text normalization: stemming, lemmatization
- dependencies and dependency parsing
- feature engineering for NLP tasks
- introduction to supervised machine-learning
- visualizing data
- evaluating machine learning models

### Initial competences

Basic knowledge about NLP techniques.

Basic programming skills with Python, such as:

1. Having general knowledge about how computer programs work
2. Having the practical knowledge and skills to develop simple computer programs using Python
3. Capacity to break down an assignment into smaller subtasks
4. Ability to find and correct bugs in code

The initial competences can be met by following the courses Introduction to Processing Language with Python - A704064 (1<sup>st</sup> semester) and Natural Language Processing - A704066

(2<sup>nd</sup> semester).

### Final competences

- 1 Having the practical knowledge and skills to integrate NLP libraries in Python code
- 2 Having the practical knowledge and skills to build machine-learning models and to evaluate their performance.
- 3 Ability to apply NLP tools and machine-learning skills to large-scale programming projects

### Conditions for credit contract

This course unit cannot be taken via a credit contract

### Conditions for exam contract

This course unit cannot be taken via an exam contract

### Teaching methods

Practicum, self-reliant study activities

### Learning materials and price

Handouts and materials for download on Ufora.

Students should have a laptop and bring it with them to the class.

Estimated total price of learning material: 0 €

### References

- Python Software Foundation. *Official Python documentation*. <http://www.python.org/doc/>
- Aurélien Géron: *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (1st ed.), O'ReillyMedia, 2017.
- Steven Bird, Ewan Klein, & Edward Loper. *Natural Language Processing with Python. Analyzing Text with the Natural Language Toolkit*. <http://www.nltk.org/book>
- D. Jurafsky, J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics* (2nd ed.), Prentice-Hall, 2009.

### Course content-related study coaching

Discussion forum on Ufora

Possibility to contact lecturers via e-mail

### Evaluation methods

continuous assessment

### Examination methods in case of periodic evaluation during the first examination period

### Examination methods in case of periodic evaluation during the second examination period

### Examination methods in case of permanent evaluation

Assignment

### Possibilities of retake in case of permanent evaluation

examination during the second examination period is possible

### Extra information on the examination methods

In order to pass, students must hand in at least 80% of all assignments given after each session.

The final assignment consists of a coding project that the student defines in consultation with the teacher. Every student should be able to explain the full code of the end result of the project.

Second exam opportunity:

Assignment: same final assignment; students submit a new, improved version.

### Calculation of the examination mark

Assignment (100%) [see the exception as described in Details Evaluation]

### Facilities for Working Students

Class attendance is strongly recommended.

Limited possibility of feedback via e-mail, restricted to answering specific questions