NAME OF THE COURSE	**Introduction to natural language processing**					
Code	PMII40	Year of study	GU-1			
Course teacher	doc.dr. sc. Branko Žitko	Credits (ECTS)	5,0			
Associate teachers		Type of instruction (number of hours)	L	S	E	F
			30		30	
Status of the course	obligatory	Percentage of application of e-learning				
COURSE DESCRIPTION						
Course objectives	Introduction to natural language processing. Processing of the morphological, syntactic, semantic and pragmatic level from the linguistic and computer perspective. Learning the basic models and algorithms for natural language processing.					
Course enrolment requirements and entry competences required for the course	Competences: linear algebra, probability, object-oriented programming in Python.					
Learning outcomes expected at the level of the course (4 to 10 learning outcomes)	describe methods of processing natural language prepare data for natural language processing implement algorithms for natural language processing in Python evaluate and compare the results of natural language processing					
Course content broken down in detail by weekly class schedule (syllabus)	Week 1: Lectures: Introductory lecture: teachers, student obligations, elements of monitoring, examination, evaluation, presentation of course objectives, literature Problems of natural language processing - weighted categorization Exercise: Working with text files and Web content in Python Week2: Lectures: Basics of language processing, regular expressions, words tokenization, words normalizations and stemming, sentence segmentation and decision trees Exercise: Regular expressions in Python, segmentation of words and sentences using regular expressions, extract text from Web pages Week3: Lectures: Edit distance of two strings, calculation of the minimum distance, backtrace in calculating alignment, weighted edit distance, edit distance in computational biology Exercise: Implementing edit distance in Python, implementation of backtrace in calculating edit distance Week4: Lecture: Language modelling, n-grams, estimate the probability of n-grams, evaluation and perplexion, generalization and zero problem, Laplace smoothing, interpolation and backoff, good-touring smoothing Exercises: Modelling of n-grams in Python, generating a language based on n-grams. Week5: Lectures: Correcting spelling errors, noise channel, bugs in actual words Exercise: Implementing algorithm for correcting spelling errors on the basis of dictionaries and edit distance Week6:					

	Lectures: Text classification, naive Bayes classifier, learning the classifier, relationship with the language model, multinomial naive Bayes classifier, precision and recall, F-measure, evaluation of classifiers Practices: Implementation of the naive Bayes classifier in Python Week7: Lectures: Sentimental analysis and basic algorithms, sentiment lexicons, learning lexicon Exercise: Implementing sentimental analyzer in Python based on the lexicon Week8: Lectures: discriminative and generative classifiers, attributes of discriminative classifier, linear classifier based on text features, maximum entropy model, optimization of probability Exercises: Implementing feature based classifier in Python Week9: Lectures: Named Entity Recognition (NER), information extraction, evaluation of NER, sequential model for NER, maximum entropy sequence model Exercise: Implementing maximum entropy Markov model for identifying names of people Week10: Lectures: Relation extraction (RE), the use of samples with RE, supervised, semisupervised and unsupervised RE Exercise: Training of maximum entropy Markov model Week11: Lectures: Parse text, syntactic structures, empirical approach to parsing, exponential problem of parsing Exercises: Modeling CFG Week12: Lectures: Probabilistic parsing, context-free grammar (CFG) and probabilistic context-free grammar (PCFG), transformation grammar, CKY parsing Exercise: Implementing CKY parser for PCFG Week13: Lectures: Information retrieval IR, term-document matrix, inverted indexes, query processing based on the inverted index, phrasal queries and positional indices, IR-ranking, scoring, TF-IDF weight vector space model Practices: Implementation of IR systems based on TF-IDF Week14: Lectures: Systems for question answering (QA), answer types and formulating queries, answer extraction, the use of knowledge in QA Practices: Implementation of QA systems using Wiki resources Week15: Lectures: Introduction to summarization, generating summaries and answer extraction, evaluation of summarization, summary of a set of documents Exercises: implementation of summarization based on information extraction	
Format of instruction	☒ lectures ☐ seminars and workshops ☒ exercises ☐ on line in entirety ☐ partial e-learning ☐ field work	☒ independent assignments ☐ multimedia ☐ laboratory ☐ work with mentor ☐ homework assignments
Student responsibilities	Class attendance, active participation in the learning process, colloquiums, oral exam	

<table>
<tr><td>Screening student work <i>(name the proportion of ECTS credits for each activity so that the total number of ECTS credits is equal to the ECTS value of the course)</i></td><td><table><tr><th>Name</th><th>Ects</th><th>Name</th><th>Ects</th><th>Name</th><th>Ects</th></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table></td></tr>
<tr><td>Grading and evaluating student work in class and at the final exam</td><td>Activity of students in lectures and exercises (attendance, problem solving, general activity in the classroom) (50%).
Written exam (50%)
The final grade is derived on the basis of all the above ratings.</td></tr>
<tr><td>Required literature (available in the library and via other media)</td><td><table><tr><th>Title</th><th>Number of copies in the library</th><th>Availability via other media</th></tr><tr><td>D. Jurafsky, J. H. Martin, (2000) Speech and Language Processing, PrenticeHall</td><td>1</td><td></td></tr></table></td></tr>
<tr><td>Optional literature (at the time of submission of study programme proposal)</td><td>S. Bird, E. Klain, E. Looper, (2009) Natural Language Processing with Python, O'Reilly Media</td></tr>
<tr><td>Quality assurance methods that ensure the acquisition of exit competences</td><td>talk with students
student evaluation using the anonymous survey
the success of students in the exam
self-assessment.</td></tr>
<tr><td>Other (as the proposer wishes to add)</td><td></td></tr>
</table>