

Short Title:	Programming for NLP APPROVED
Full Title:	Programming for Natural Language Processing
Module Code:	MHLT H6019
ECTS credits:	10
NFQ Level:	9
Module Delivered in	1 programme(s)
Module Contributor:	Irene Murtagh
Module Description:	To provide students with in-depth knowledge and skills of the role of programming skills and strategies for NLP contexts, To give students the skills necessary a wide range of NLP related algorithmic techniques for software development, To provide students with the necessary theoretical and practical applications framework for programming natural language applications, To provide students with in-depth knowledge and skills regarding the complex challenges of processing human languages in software, To instil an in-depth appreciation of the importance of quality in software development.
Learning Outcomes:	
<i>On successful completion of this module the learner will be able to</i>	
<ol style="list-style-type: none"> 1. Develop programmes that can manipulate and analyse human language data. 2. Apply core knowledge concepts from NLP and linguistics to analyse and describe human language. 3. Apply data structures and algorithms in NLP. 4. Store complex language data in standard formats. 5. Evaluate the performance and effectiveness of NLP techniques. 	

Module Content & Assessment

Indicative Content
Language Processing and Python (10%) Computing with human language as texts, words, tokens and symbols. Texts as lists of words. Control structures and decision making. Automatic natural language understanding
Accessing digital lexical resources and text corpora (10%) Accessing text corpora. Conditional frequency distributions. Code reusability as a strategy Lexical resources. WordNet and FrameNet
Processing raw text (10%) Accessing text from the Internet. Strings and text processing. Unicode representation. Applying regular expressions for detecting word patterns in syntax. Normalising text. Tokenizing text. Segmentation. Processing lists and strings
Writing software code for NLP (10%) Algorithm design. Programming style for good quality code. Structured programming. Functions and methods. Programme development strategies
Categorising and tagging words (10%) Using a Tagger. Tagged corpora. Mapping words to properties using Python dictionaries. Automatic tagging. N-Gram tagging. Determining the lexical category of a word
Extracting information from digital text (10%) Information extraction. Chunking. Developing and evaluating chunkers. Recursion in linguistic structure. Named entity recognition. Relation extraction
Analysing the sentence in code (10%) Modelling of linguistic patterns. The role of linguistic syntax. Context-free and context-sensitive grammar. Parsing with context-free grammar. Argument structure, valence and dependency grammar. Grammar development
Building Feature-Based Grammars (10%) Grammatical features and attribute value matrices. Processing feature structures. Feature-based grammars
Analysing the meaning of sentences (10%) Natural language understanding challenges. Propositional logic. First-order logic. The semantics of English sentences. Discourse semantics. Description logics
Current issues in managing linguistic data in software (10%) The Life Cycle of a Corpus. Corpus Structure. Acquiring Data. Working with XML. Working with Toolbox Data. Describing Language Resources Using Open Language Archives Community (OLAC) metadata

Indicative Assessment Breakdown	%
Course Work Assessment %	100.00%

Course Work Assessment %				
<i>Assessment Type</i>	<i>Assessment Description</i>	<i>Outcome addressed</i>	<i>% of total</i>	<i>Assessment Date</i>
Practical/Skills Evaluation	Practical work based on lecture material	1,5	20.00	Every Week
Project	The student will typically be expected to construct effective structured code that manipulates and analyses human language data, along with the appropriate selection of data structure and algorithm choices.	1,2,3	40.00	Week 6
Project	The student will typically undertake work in creating and manipulating complex language data in a variety of formats within code and be expected to generate metrics that evaluate the performance and effectiveness of the NLP techniques and choices made.	4	40.00	Week 9

No Final Exam Assessment %

Indicative Reassessment Requirement
Coursework Only <i>This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.</i>

ITB reserves the right to alter the nature and timings of assessment

Indicative Module Workload & Resources

Indicative Workload: Full Time	
Frequency	Indicative Average Weekly Learner Workload
Every Week	24.00
Every Week	24.00
Every Week	152.00
Indicative Workload: Part Time	
Frequency	Indicative Average Weekly Learner Workload
Every Week	24.00
Every Week	24.00
Every Week	152.00

Resources
<i>Recommended Book Resources</i>
<p>Steven Bird, Ewan Klein, and Edward Loper 2009, <i>Natural language processing with Python</i>, O'Reilly Sebastopol, Calif. [ISBN: 0596516495]</p> <p><i>XML Processing with Perl, Python, and PHP: Also Covers Tel, Rebol, Ruby, and AppleScript</i>, John Wiley & Sons Hoboken [ISBN: 0782140211]</p> <p>Daniel Jurafsky, James H. Martin, <i>Speech and language processing</i>, Upper Saddle River, NJ Pearson 2009 [ISBN: 0135041961]</p>
<i>Supplementary Book Resources</i>
<p>Nolan, Brian and Carlos Periñán [Studies in language Companion Series 150]., <i>Language processing and grammars: The role of functionally oriented computational models.</i>, John Benjamins Publishing Company. Amsterdam and New York</p> <p>Xuedong Huang, Alex Acero, Hsiao-Wuen Hon 2001, <i>Spoken language processing</i>, Prentice Hall PTR Upper Saddle River, N.J. [ISBN: 0130226165]</p> <p>David Harrel 2012, <i>Algorithmics</i>, Boston, Mass. ; Addison-Wesley, 2004. [ISBN: 0321117840]</p> <p>Grant S. Ingersoll, Thomas S. Morton, Andrew L. Farris, <i>Taming Text</i>, Manning Publications [ISBN: 193398838X]</p> <p>Jeff McNeil, <i>Python 2.6 Text Processing Beginners Guide</i>, Packt Publishing [ISBN: 1849512124]</p> <p>David Mertz 2003, <i>Text processing in Python</i>, Addison-Wesley Boston [ISBN: 0321112547]</p> <p>Nolan, Brian and Elke Diedrichsen. Studies in language Companion Series 145]. 2013, '<i>Linking Constructions into functional linguistics – The role of constructions in RRG grammars</i>', John Benjamins Publishing Company. Amsterdam and New York</p>
<i>This module does not have any article/paper resources</i>
<i>Other Resources</i>
<p>Website: CALLHOME German Lexicon. http://www.language-archives.org/item/oa_i:www ldc.upenn.edu:LDC97L18</p> <p>Website: MULTILEX multilingual lexicon. http://www.language-archives.org/item/oa_i:el ra.icp.inpg.fr:M0001</p> <p>Website: NLTK http://www.nltk.org/</p> <p>Website: OLAC http://www.language-archives.org/</p> <p>Website: Python NLP code http://gnosis.cx/TPiP</p> <p>Website: Python Programming http://www.python.org/</p> <p>Website: Slex Siemens Phonetic lexicon http://www.language-archives.org/item/oa_i:elra .icp.inpg.fr:S0048</p>

Module Delivered in

Programme Code	Programme	Semester	Delivery
BN_KMHLT_R	Master of Science in Computing in Multimodal Human Language Technology	1	Mandatory