Syllabus

Introduction to Computational Linguistics

Course Description

The course "Introduction to Computational Linguistics" centers on the basic and fundamental concepts of this interdisciplinary area of Linguistics, Informatics and Cognitive Science. Our purpose is to cover a wide range of theoretical and technical issues from Speech Recognition and Synthesis to Natural Language Processing and Machine Translation. Significant topics from Semantics, Syntax, Morphology and Phonology will be introduced through the dynamic prism of several state-of-the-art computing tools, applications, models and theories. This course will use a methodology of empirical linguistic analysis and processing of natural language that includes regular expressions, language modeling, machine learning, morphological / syntactical parsing, and semantic analysis and representation.

In particular, the main aim of the course is to familiarize students with significant and on-going research questions and theoretical approaches in this field and to provide them access to various tools and applications, while at the same time introducing them to language coding through programming. Moreover, we will also focus on how linguistic theory is applied to the most up-to-date text processing techniques, word meaning and semantic interpretations. Theoretical and technical issues such as n-grams models, neighborhood density, Context-free Grammars, morphosyntactic tagging, vector semantics, computing with word senses will be supported by exercises and mini-projects that will enable students to use practice tools, corpora and apply various semantic algorithms.

Course Objectives

Upon completion of this course, students should be able to:

- *understand* basic concepts of Computational Linguistics.
- *follow* the current trends of an ever-evolving scientific area.
- *recognize* mainstream linguistic theories in a more technical environment.
- (computationally) *analyze* the Greek language on different levels.
- *acquire* theoretical and computational skills in language processing.
- *interpret* various phenomena by approaching them computationally.
- *get* stimuli and motives for further studying this area.
- "*dirt*" their hands by trying some basic and preliminary programsming.

General Skills

- Implementing theory in practice
- Investigate, analyze and synthesize data and information, using the necessary technologies
- Decision making
- Individual or team work
- Exercise criticism and evaluation
- Promoting free, creative and inductive thinking

Course Material and EduTech

- Notes and on-line available articles
- Presentations material
- Multimedia material
- Interactive quizzes and exercises
- Computational tools and applications
- Python programming tutorial (mock lab)
- Educational Technology for Teaching, Tutoring and Communicating with the students

Students Evaluation

(1.) 10% Participation [Bonus]

This interactive course blends the study of theoretical subjects (e.g., syntax, morphology, semantics, phonology) with more technical and computational issues (e.g., NLTK toolkit, Python, Wordnet/ Framenet, Hidden Markov Models, etc.). This course is designed in an attempt to reflect this mix in the schedule of reading assignments, activities, and the like. In the following course schedule, I present a mixed and balanced use of reading material throughout the course to ensure the successful completion of the exercises (2nd type of assessment), the short essay/ mini project (3rd type of evaluation) and the well-structure preparation for the final examination which are directly dependent on active participation in the lesson and reading the weekly material. Be aware that class lectures and discussions are designed to supplement (not supplant!) the material.

(2.) 30 % Assignments (multiple-choices, quizzes, article reviewing) [Optional]

There will be a group of assignments/ exercises that will be covered in their majority during the course. Each exercise will follow a significant subject (and related references), a discussion, a sample exercise and / or a small-range tutorial. Each type of exercise will be based on the quick and (often) direct-evaluation of the students, as well as the preparation for corresponding exercises in the final examination.

(3.) 70 % Short Essay/ Mini-project [instead of exams]

There will also be an opportunity for individual or group work, which will be presented (in 10 minutes) during the last week of the course. The nature of essay/ project will determine whether it will be individual or group work. Students will also have an opportunity to evaluate and present specific articles on the matter and drawing examples from the Engish and Greek language, as well as the ability to use and evaluate specific tools and applications. Finally, there will also be a group work option for a basic Python programming project.

(4.) 70% Exams

The final evaluation of the students is one final exam that will naturally draw on knowledge accumulated in the course up to that point and the main handbook; it will be roughly drafted around a similar format: a series of multiple choices and short-answer, identification questions, discussions, apps evaluation, and an essay question concerning specific computational issues and theories.

(5.) 10% Python online Tutorial [bonus]

References

Coursebook

Jurafsky and Martin (2000, 2007, 2011). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (1st, 2nd, 3rd edition). Prentice Hall. (https://web.stanford.edu/~jurafsky/slp3/)

Secondary BIbliography

Roark B. & Sproat R. (2007). *Computational Approaches to Morphology and Syntax*. Oxford: Oxford University Press.

Manning & Schütze (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Bird S., Klein E. & Loper E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* O' Reilly Media.

Καρασίμος, Α. (2011). *Υπολογιστική Επεξεργασία της Αλλομορφίας στην Παραγωγή Λέξεων της Νέας Ελληνικής*. Διδακτορική Διατριβή. σσ. 305. Πανεπιστήμιο Πατρών: Τμήμα Φιλολογίας. DOI: 10.13140/RG.2.1.1570.7926

Supplementary Bibliography

Basirat A., Faili H. & Nivre J. (2015). A statistical model for grammar mapping. *Natural Language Engineering 22* (2): 215–255.

Goldsmith, J. (2000). Linguistica: An Automatic Morphological Analyzer.The Proceedings from the Main Session of the Chicago Linguistic Society's Thirty-sixth Meeting. Arika Okrent and John Boyle (eds.) 1-36.

Hammarström, H. and Borin, L. (2011). Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Maletti A. (2017). Survey:Finite-state technology in natural language processing. *Theoretical Computer Science 679*, pp. 2–17.

(*a list of more than 30 supplementary but important articles/ resources plus several links from tools and applications that will be presented during the course teaching*)

Weekly Schedule

Εβδομάδα	**Τίτλος**	**Στόχοι**	**Βιβλιογραφία**	
1	Course and Historical Overview	• Course Overview • Historical Overview • Computers in Linguistics and Natural Language Processing	Jurafsky & Martin (2017), Chapter 1	
2	Regular Expressions	• Regular Expressions • Pattern matching • Text normality	Jurafsky & Martin (2017), Chapter 2	
3.	Regular Languages & Finite-State Automata	• Regular Languages • Finite-State Theory • Transducers & automata • Two-Level Morphology (PC-KIMMO)	Jurafsky & Martin (2017), Chapter 2 & 3	
4.	Language Modelling with N-Grams	• Language Modelling • N-Grams	Jurafsky & Martin (2017), Chapter 4	Codacademy Codeschool
5.	Tagging and Hidden Markov Models	• POS Tagging • Tagging • Word Classes & Tagsets • Hidden Markov Models	Jurafsky & Martin (2017), Chapter 5 & 6	NLTK Toolkit Python Tutorial I
6.	Parsing	• Syntactic Parsing • Morphological Parsing • Semantic Parsing • Statistical Parsing	Jurafsky & Martin (2017), Chapter 12, 13 & 14	NLTK Toolkit Python Tutorial II
7.	Context-Free Grammars	• Semantic Ambiguity • Context-Free Grammars • Grammar mapping	Jurafsky & Martin (2017), Chapter 11	

8.	Machine Learning	• Machine Learning • Morphological Learning • Unsupervised and Supervised Techniques	Καρασίμος (2011), Κεφάλαιο 4	Linguistica
9.	Word Meaning and Vector Semantics	• Word Meaning • WordNet & FrameNet • Information Retrieval and Extraction • Vector Semantics	Jurafsky & Martin (2017), Chapter 17 & 20	
10.	Computational Semantics & Ontologies	• Computational Semantics • Semantic Representation • Thesauri and Ontologies	Jurafsky & Martin (2017), Chapter 21	
11.	QA Systems & Dialogue Systems	• QA Systems • Dialog Systems • Summarization	Jurafsky & Martin (2017), Chapter 27, 28, 29 & 30	
12.	Machine Translation	• Machine Translation • Translation tools and techniques	Jurafsky & Martin (2017), Chapter 26	
13.	Revision and Recap	• Recap of key concepts • Mock Exam		