Course Title (in English)	Introduction to Natural Language Processing
Course Title (in Russian)	Введение в обработку естественного языка
Lead Instructor	Panchenko, Alexander
Contact Person	Alexander Panchenko
Contact Person's E-mail	a.panchenko@skoltech.ru

Course Description

This course gives introduction to the field of Natural Language Processing (NLP). The course provides a panorama of various NLP tasks and applications such as part-of-speech tagging, named entity recognition, and word sense disambiguation.
The course is largely based on the classic textbook by Jurafsky&Martin, but also features material on graph-based models for NLP and data annotation / crowdsourcing for NLP.

The course on "Deep Learning for Natural Language Processing" at Skoltech provides complementary material to this introductory course focusing more on modern neural models and approaches, e.g. Transformers, and less on the various applications and related topics, such as crowdsourcing, graph-based NLP, and dependency parsing.

Course Prerequisites / Recommendations	Required: - General computer science on BA-level - Python programming language Advantageous: - introductory knowledge of machine learning - introductory knowledge of statistics

Аннотация

Данный курс представляет собой введение в область обработки естественного языка (NLP). В рамках курса рассматриваются различные задачи и приложения в области NLP, такие как морфологический анализ, распознавание именованных сущностей и устранение неоднозначности смысла слов. Курс в значительной степени основан на классическом учебнике Jurafsky&Martin, но также содержит материалы по графовым моделям для NLP и аннотации данных/краудсорсингу лингвистических данных.

Курс «Deep Learning for Natural Language Processing» в Сколтехе дополняет материал из данного курса и в большей степени фокусируется на современных нейронных моделях и подходам, таких как Transformer и не покрывает различные приложения и смежные темы, такие как краудсорсинг, графовые методы и синтаксический парсинг.

Course Academic Level	Master-level course suitable for PhD students
Number of ECTS credits	3

Topic	Summary of Topic	Lectures (# of hours)	Seminars (# of hours)	Labs (# of hours)
Introduction into Natural Language Processing. Applications	Introduction to the field of Natural Language Processing (NLP). Panorama of NLP tasks. Dialogue systems as a prominent application of NLP.	3	3	0
Distributional Semantics and Word Sense Disambiguation	Sparse count-based vector spaces. Word Embeddings, Word2Vec, GloVe and related models. WordNet, Word Sense Disambiguation, Word Sense Induction.	3	3	0
Sequence Tagging	Sequence tagging task and its applications such as Named Entity Recognition. Conditional Random Field and related models.	3	3	0
Language Models and Machine Translation	Language Models: unigram/bigram/n-gram, Markov Chain, Zipf's law, Hidden Markov Models. Statistical and neural models for machine translation.	3	3	0
Syntactic parsing	An overview of various types of syntactic parsing common in NLP: chunking, dependency parsing, constituency parsing and others. Methods for dependency parsing.	3	3	0
Graphs for NLP: Linguistic Networks and Clustering	Motivation for graph representation in NLP. Types of graphs in NLP tasks. Graph Clustering, Chinese Whispers, Markov Clustering, and related algorithms. Applications.	3	3	0
Data Annotation and Crowdsourcing for NLP	The majority of the successful NLP models rely on linguistically annotated data in one form or another. Often in practical applications for a given language and domain such a dataset is not available not making it possible to apply the supervised models. In this lecture you will learn how to setup creating of the required data.	3	3	0

Assignment Type	Assignment Summary
Homework Assignments	Word sense induction (WSI). The goal of this task is to apply distributional models, such as word2vec, to discriminate between different meanings of a single word e.g. python as programming language and python as a snake. Towards this goal, clustering of semantic vectors will be explored. The solutions will be submitted to the CodaLab platform with a public leaderboard. A Jupyter notebook with code and results of experiments shall be provided.
Homework Assignments	Taxonomy enrichment. The goal of this task is to construct a populate / linguistic tree automatically using distributional models of word meaning. The solutions will be submitted to the CodaLab platform with a public leaderboard. A Jupyter notebook with code and results of experiments shall be provided.
Homework Assignments	Semantic role labelling (SRL). The goal of this task is to label text with semantic roles indicating meaning of individual parts. More specifically, this SRL task will be casted as a sequence tagging problem. We will consider the domain comparative argument mining: parsing texts which compare two products with respect to some aspect, e.g. comparing what is better Python or Matlab for NLP? The solutions will be submitted to the CodaLab platform with a public leaderboard. A Jupyter notebook with code and results of experiments shall be provided.
Test/Quiz	Answer questions on the materials of each lecture.

Type of Assessment	Graded

Grade Structure

Activity Type	Activity weight, %
Test/Quiz	25
Homework Assignments	75

A:	86
B:	76
C:	66
D:	56
E:	46
F:	0
Attendance Requirements	Mandatory with Exceptions

Maximum Number of Students

	Maximum Number of Students
Overall:	35
Per Group (for seminars and labs):	

Course Stream	Science, Technology and Engineering (STE)
Course Term (in context of Academic Year)	Term 2
Course Delivery Frequency	Every year

Students of Which Programs do You Recommend to Consider this Course as an Elective?

Masters Programs	PhD Programs
Advanced Computational Science Data Science Engineering Systems Information Science and Technology Internet of Things and Wireless Technologies Space and Engineering Systems	Computational and Data Science and Engineering Engineering Systems Life Sciences Mathematics and Mechanics

Course Tags

Programming
Engineering

Required Textbooks	ISBN-13 (or ISBN-10)
Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed.). Online. https://web.stanford.edu/~jurafsky/slp3/	

Recommended Textbooks	ISBN-13 (or ISBN-10)
Manning, C. and Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA.	
Manning, C. D., Raghavan, P., and Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.	

Web-resources (links)	Description
https://web.stanford.edu/~jurafsky/slp3/	Textbook online

Equipment
Access to a computational server with GPUs (e.g. Nvidia 2080 Ti or similar) may be useful but the Google CodaLab platform should be normally sufficient.

Software
Python 3

Knowledge
Methods for natural language processing
Methods for evaluation of natural language processing systems
Software libraries and frameworks for writing programs for natural language processing
Resources of research literature for natural language processing

Skill
Select the proper language models and computational methods for a variety of NLP tasks.
Analyze and evaluate statistical NLP methods in various applications.
Develop statistical models and methods for various NLP applications.
Conduct methodological research in natural language processing.

Experience
Implementation of a range of computational methods and linguistic models for various NLP-problems.

Select Assignment 1 Type	Homework Assignments
Or Upload Example(s) of Assignment 1	https://ucarecdn.com/050e5d94-c913-4e1a-8375-77f3fc4ee4b0/
Assessment Criteria for Assignment 1	- Results (improved over baseline, top1 solution, top 20% solution among class) - Code (readability and reproducibility) - Report (methodology and discussion of results)
Select Assignment 2 Type	Test/Quiz
Input Example(s) of Assignment 2 (preferable)	

How many parameters would have a 3-gram language model with a vocabulary of 100 characters?
- 300
- 10 thousand
- 1 million
- 1 billion

Which architecture for language models will probably consume more memory while training with long sequences?
- n-gram language model
- convolutional neural network
- recurrent neural network
- transformer neural network

Which architecture for language models will probably be the slowest to train with long sequences (if one has a GPU)?
- n-gram language model
- convolutional neural network
- recurrent neural network
- transformer neural network

Which loss function for language models is NOT equivalent to all the others?
- cross-entropy
- mean squared error
- log likelihood
- perplexity

Which parameter was roughly the same for GPT, GPT-2 and GPT-3?
- number of layers
- size of the training corpus
- dimensionality of hidden state
- vocabulary size

With which task it would be difficult to create a prompt for GPT zero-shot application?
- Filling gaps in a text with appropriate words or phrases
- Text classification (e.g. by sentiment)
- Reading comprehension (answering a question for a text)
- Machine translation

Which parameter is meaningful with greedy (non-probabilistic) text generation?
- temperature
- top k
- repetition penalty
- top P

Which statement is NOT a reason why generation with positive lexical constaints is so difficult?
- Typically, a language model has not been trained to condition on anything except the already generated text.
- By simply forcing the model to generate certain words, we may prevent it from generating a fluent text.
- Because text generation is autoregressive, the model cannot naturally "plan" in advance to generate a text with certain words.
- During training, the model may have never seen a text when the given words have occured together.

Which approach to controllable text generation is the most resource-intensive during training?
- Gradient-based prompt tuning
- Model fine-tuning with reinforcement learning
- PPLM (steerable lanugage models)
- Model fine-tuning conditional on control codes

Which approach to controllable text generation is the most resource-intensive during inference?
- Gradient-based prompt tuning
- Model fine-tuning with reinforcement learning
- PPLM (steerable lanugage models)
- Model fine-tuning conditional on control codes

Which approach to controllable text generation does not require computing gradients of the model?
- Prompt tuning
- PPLM (steerable lanugage models)
- GeDi (generative discriminators)
- Model fine-tuning conditional on control codes

Assessment Criteria for Assignment 2

- Number of correctly answered questions