COURSE DESCRIPTION CARD - SYLLABUS

Course name
Advanced Natural Languages Processing

Course

Field of study
Computing

Year/Semester
1/2

Area of study (specialization)
Artificial Intelligence

Profile of study
general academic

Level of study
Second-cycle studies

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
45

Laboratory classes
15

Other (e.g. online)

Tutorials

Projects/seminars

Number of credit points

5

Lecturers

Responsible for the course/lecturer:
Mateusz Lango
email: mateusz.lango@cs.put.poznan.pl
tel. 61 665 21 24
Faculty of Computing and Telecomunications
Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

A student starting this course should have basic knowledge of probability and statistics (normal, binomial and Bernoulli distributions, maximum likelihood estimation, unbiased, consistent, effective estimators), as well as in-depth knowledge of machine learning (ensembles, k-NN, Naive Bayes, SVM) and deep learning (multi-layer neural networks, convolutional networks, backpropagation). Additionally, basic knowledge of text processing, equivalent to the course "Web Mining" or "Natural language processing," is also assumed (regular expressions, stemming, lemmatization, stopwords, bag-of-words model, measures of text similarity).

The student should have the ability to solve basic math problems regarding probability and statistics, should be proficient in Python (with a deep learning library like PyTorch or TensorFlow), and should know how to obtain information from indicated sources.

In terms of social competencies, the student must understand that in computer science knowledge and skills quickly become obsolete. The student should present attitudes such as honesty, responsibility, perseverance, cognitive curiosity, creativity, and respect for other peoples.

Course objective

The aim of the course is to familiarize students with the methodology, resources and tools used in natural language processing. Lectures focus on the discussion of classical statistical methods as well as modern techniques based on the new achievements of deep learning for problems such as automatic translation, sentiment analysis, text classification, dialogue systems, named entity recognition, syntax analysis, and topic modeling. The additional goal of the course is to develop the ability to analyze statistical and machine learning models in various respects (computational complexity, type of training data and required sample size, model assumptions / limitations, inference methods) and their use to solve non-trivial problems regarding text data.

Course-related learning outcomes

Knowledge

1. Student has advanced and in-depth knowledge of the construction of computer systems that process natural language with statistical methods - [K2st_W3]

2. Student has an in-depth understanding of the architectures of deep neural networks used in human language technologies (in particular recurrent and recursive architectures) - [K2st_W3]

3. Student has advanced and in-depth knowledge related to selected issues, such as language modeling, syntax analysis, distributional semantics, named-entity recognition, machine translation, dialog agents - [K2st_W3]

4. Student knows development trends and the essential new achievements of natural language processing (including modern deep machine learning architectures) – [K2st_W4]

5. Student knows advanced methods, techniques, and tools used in the construction of dialogue systems, translators, parsers, and question answering systems - [K2st_W6]

6. Student understands advanced methods used in research in the field of natural language processing - [K2st_W6]

Skills

1. Student is able to obtain information on natural language processing techniques from literature and other sources (in Polish and English), integrate them, interpret and critically evaluate them, draw conclusions and justify opinions - [K2st_U1]

2. Student is able to obtain appropriate data sets for particular natural language processing tasks (e.g. from the CLARIN database) - [K2st_U1]

3. Student is able to plan and carry out computational experiments on text data, interpret the obtained results and draw conclusions - [K2st_U3]

4. Student - when formulating and solving engineering tasks - integrates knowledge from various areas of machine learning, software engineering, natural language processing and linguistics. - [K2st_U5]

5. Student can assess the usefulness and the possibility of using new achievements of machine learning to solve problems in linguistic engineering - [K2st_U6]

6. Student can determine the directions of a further self-study - in particular for learning new techniques of state-of-the-art natural language processing - [K2st_U16]

Social competences

1. Student understands that in human language technologies, knowledge and skills become obsolete very quickly - [K2st_K1]

2. Student understands the importance of using the latest achievements in the field of human language technologies in solving research and application problems - [K2st_K2]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

a) Verification of the assumed learning outcomes of the lectures is carried out by:

- assessment of knowledge and skills demonstrated in a written exam containing open-ended and multiple-choice questions,
- discussion of the test results.

b) The skills and knowledge acquired by the student during the laboratories will be assessed by

- evaluation of problem sets, including simple implementation tasks in Python (requiring the execution of experiments as well as the analysis and interpretation of the obtained results),
- assessment of the presentation prepared by the student, which will discuss a selected issue in NLP.

Student can obtain additional points for activity during classes, especially for:

- discussing methods/problems/approaches in NLP which are beyond the scope of the course e.g., through short presentations,
- remarks related to the improvement of teaching materials,
- identifying students' perceptual difficulties enabling ongoing improvement of the teaching process.

The following grading scale is used for both lectures and tutorials: above 51% of points - satisfactory (3.0), 61% - satisfactory plus (3.5), 71% - good (4.0), 81% - good plus (4.5), 91% - very good (5.0).

Programme content

1. Introduction to natural language processing. Language as a system: an attempt to define language, the duality of patterning, language variability in synchronous view, linguistic relativity (Sapir–Whorf hypothesis), universalist theories. A brief historical overview of research in linguistics and natural language processing and its impact on artificial intelligence development. The characteristics of text data: the notion of a corpus, out-of-vocabulary words, language ambiguity. Applications of natural language processing.

2. Statistical language modeling. N-gram models: maximum likelihood estimation, linear interpolation of n-gram models, bucket method, smoothing methods, Katz back-off model, and general outline of the Knesser-Ney model. Advanced language models: class n-gram model, Brown semantic clustering, semantic dependencies in clustering dendrogram. Evaluation of language models.

3. Neural language models. Log-linear models. Neural autoregressive 3-gram model, autoregressive model with embedding matrix. The problem of scaling neural models to large dictionaries and solutions: importance sampling and hierarchical softmax.

4. Text classification. Continous Bag-of-words, bag-of-character-grams, bag-of-classes, bag-of-ngrams representations. Convolutional neural networks for text classification: 1D convolution layer (on characters and words), pooling-over-time, the idea of multiple channels in the context of distributed representation. Case study: sentiment classification: classic unsupervised approaches, Ossgood's sentiment model, the problem of negation, sentiment lexicons, an example text processing workflow for sentiment classification of short user utterances in Twitter.

5. Semantics. Semantic relations and their use in the construction of computer lexicons: antonymy, homonymy, synonymy, polysemy, homonymy, hyponymy, hyperonymy. Polish WordNet. The meaning of words and their distributional properties. Word-context matrix, positive pointwise mutual information (PPMI). Distributed representations of words: global methods (HAL).

6. Distributional semantics. Construction of words embeddings: iterative methods (word2vec), methods for languages with rich morphology (FastText). Negative sampling. Semantic and syntactic analogies. Theoretical explanation of their formation. The problem of out-of-vocabulary words, the problem of polysemy, the problem of social bias in word representations.

7. Introduction to sequence prediction. Part of speech (PoS) and named-entity recognition (NER): problem definition, evaluation methods, and different encoding schema (BIO, IOB, etc.). The comparison of structure prediction with classification.

8. Statistical models for sequence prediction. 3-gram hidden Markov models (Trigram HMMs), model estimation, Viterbi algorithm. Maximum-entropy Markov models (MEMM). Label bias problem. Feature engineering for NER and PoS problems.

9. Conditional random fields for sequence prediction (linear-chain CRF). Probabilistic graphical models - directed and undirected Markov graphs (revision): factors graph. Backward-forward algorithm for CRF. Training CRF models with stochastic gradient descent algorithm.

10. Neural sequence prediction. Recurrent neural networks (Elman and Jordan architectures) using distributed representations, review of GRU and LSTM neurons, CRF layer, bidirectional models, backpropagation through time algorithm. Several tricks for the speed-up of recursive neural networks.

11. Syntax. Derivation tree, dependency tree, context-free grammars, syntactic ambiguity, probabilistic context-free grammars (definition, estimation, CKY algorithm, Chomski's normal form), introduction to lexicalized context-free probabilistic grammars. Outline of recursive neural networks.

12. Dependency parsing. Graph and transition-based methods. Shift-reduce algorithm. Chu-Liu-Edmonds algorithm. Outline of Eisner's algorithm. Universal Dependencies.

13. Machine translation. Difficulties and problems to be solved in automatic translation. Vauquois pyramid. Model IBM 1: assumptions, model construction, supervised estimation of parameters (from the corpus containing word alignments), estimation of parameters from a parallel corpus.

14. The problem of word alignment. Text clustering as a classification with latent labels. Expectation-maximization (EM) algorithm for the IBM 1 model and text clustering under Naive Bayes' assumptions. EM algorithm as alternating maximization. Introduction to phrase translation methods. Evaluation of machine translation systems (expert and automatic assessment - BLEU).

15. Neural machine translation. Encoder-decoder neural networks. Attention mechanism - various definitions, interpretation of action as automatic word alignment. Interpretability problems of attention. Language-independent distributed representations. Encoder sharing, back-translation technique. Direct optimization of BLEU: minimum risk training.

16. Transfer learning in NLP - introduction. Transfer learning from language models to text classification and other tasks: ULMFit method. Contextual word embedding: ELMO.

17. Transfer learning between languages: methods of mapping word embeddings in a supervised and unsupervised way. Orthogonal Procrustes problem and its solution through SVD decomposition. The problem of initialization of the translation matrix.

18. Transformer architecture. Problems of deep recursive networks with GRU / LSTM neurons. Quasi-recursive neural networks. The idea of multi-head attention. Positional embeddings.

19. Transfer learning in NLP. Publicly available multi-task pre-trained systems based on the transformer architecture: BERT, Universal Sentence Encoder, GPT-3, and similar. Outline of Reformer architecture.

Multilingual BERT models and the model for the Polish language: HerBERT. Limitations of modern language models. GLUE (for the English language) and KLEJ (for the Polish language) benchmarks.

20. Explaining and interpreting deep natural language processing models. Probing classifier as an example of structural methods. Behavioral methods: the idea of "challenge sets". Visualization through the highest gradient activation.

21. Review of selected natural language processing applications (selection according to students' interests): text-to-speech methods, speech recognition techniques (ASR), building knowledge graphs from texts, question answering, information retrieval (DSSM models), dialogue systems, topic modeling.

During laboratories, students perform computational experiments and implement the models discussed in the lecture. In addition, students present selected scientific papers in the form of short presentations.

Teaching methods

1. Lecture: presentation, illustrated with examples solved on the blackboard

2. Laboratory classes: computational experiments and implementation of methods presented during lectures (e.g. in Jupyter Notebook), discussion of potential problems and solutions

Bibliography

Basic

1. Jurafsky D., Martin J.H.: Speech and Language Processing, III edycja, Pearson/Prentice Hall, 2018 (access online: https://web.stanford.edu/~jurafsky/slp3/)

2. Li Deng, Yang Liu: Deep Learning in Natural Language Processing. Springer, 2018 (access through eZasoby service of PUT library)

Additional

1. Mykowiecka, A: Inżynieria lingwistyczna: komputerowe przetwarzanie tekstów w języku naturalnym, Wydawnictwo PJWSTK, 2007.

2. Yoav Goldberg, Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies), Morgan & Claypool Publishers, 2017.

3. Goodfellow I., Yoshua B., Courville A.: Deep Learning. Systemy uczące się., PWN, 2018

4. Lango M., Brzeziński D., Stefanowski J.: PUT at SemEval-2016 Task 4: The ABC of Twitter Sentiment Analysis, Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), 2016

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,0
Classes requiring direct contact with the teacher	60	3,0
Student's own work (literature studies, preparation for tests, homeworks) [1]	65	2,0

1 delete or add other activities as appropriate