COURSE DESCRIPTION CARD - SYLLABUS

Course name

Human Language Technologies (Inżynieria Lingwistyczna)

Course

Field of study

Computing

Area of study (specialization)

Inteligentne technologie informatyczne

Level of study

Second-cycle studies

Form of study

full-time

Year/Semester

2/3

Profile of study

general academic

Course offered in

Polish

Requirements

compulsory

Number of hours

Lecture

30

Laboratory classes

Other (e.g. online)

Tutorials

30

Projects/seminars

Number of credit points

4

Lecturers

Responsible for the course/lecturer:

Mateusz Lango

Faculty of Computing and Telecomunications

Piotrowo 2, 60-965 Poznań

email: mateusz.lango@cs.put.poznan.pl

tel. 61 665 21 24

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, recurrent 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 calculation problems with 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 human language technologies. Classes focus on the discussion of classical statistical methods and 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 using 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 recursive 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 - [K2st_W3]

4. Student knows development trends and the essential new achievements of linguistic engineering (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 linguistic engineering - [K2st_W6]

Skills

1. Student is able to obtain information on linguistic engineering 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 individual linguistic engineering 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 - integrate knowledge from various areas of machine learning, software engineering, 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 linguistic engineering [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 and machine learning in solving practical problems - [K2st_K2]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

The knowledge presented during the lectures will be verified by written tests containing open-ended and multiple-choice questions.

The skills and knowledge acquired by the student during the tutorials 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 HLT.

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

- discussing methods/problems/approaches in HLT which are beyond the course material, e.g., through short presentations of scientific articles,

- 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. Language as a system: an attempt to define language, the formal and semantic level of the language (signe, signifiant, signifie), double articulation of the language system, language variability, linguistic relativism, universalist theories. Selected issues in semantics: denotation, reference, connotation. Semantic relations and their use in the construction of computer lexicons: antonymy, homonymy, synonymy, polysemy, homonymy, hyponymy, hyperonymy. Polish WordNet.

2. Statistical language modeling: Markov models, 3-gram model, maximum likelihood estimation, language model evaluation, linear interpolation of 3-gram model, bucket method, smoothing, Katz back-off model, and the high-level outline of the Knesser-Ney model. The meaning of words and their distributional properties. Advanced language models: class n-gram model, Brown semantic clustering, semantic dependencies in dendrogram, neural language modeling (3-gram neural model), the problem of scaling neural models to large dictionaries (weighted sampling, hierarchical softmax). Word embedding representations: iterative methods (word2vec), global methods (HAL, GloVE), techniques for morphologically rich languages (FastText, ELMO). Semantic and syntactic analogies, the problem of out of vocabulary words, the problem of polysemy.

3. Named-entity recognition (NER) and part of speech tagging (PoS): problem definition and encoding methods (BIO, IOB). Generative models: 3-gram hidden Markov models, estimation of model parameters, Viterbi algorithm. Discriminant models: linear-chain conditional random fields (CRF), Maximum Entropy Markov Models (MEMM), backward-forward algorithm. Neural recognition of named entities: recursive neural networks (Elman and Jordan architecture) using distributed representations, review of GRU and LSTM neurons, CRF layer, bidirectional models.

4. Syntax analysis: derivation tree, dependency tree, context-free grammars, ambiguity problem, context-free probabilistic grammars (definition, estimation, CKY algorithm, Chomski's normal form), introduction to lexicalized probabilistic context-free grammars. Recursive neural networks: RecNN, backpropagation through the structure, ranking error (definition, analysis of advantages and disadvantages), recursive methods.

5. Machine translation: sources of difficulties in automating translation, Vauquois pyramid, IBM models (1 & 2), estimation of parameters from the corpus with word assignments, estimation of parameters from a parallel corpus, expectation-maximization (EM) algorithm, introduction to phrase-based machine translation. Evaluation of machine translation systems (BLEU). Neural machine translation methods:

encoder-decoder approaches, attention, language-independent distributed representations, encoder sharing, back-translation technique. Linguistic transfer.

6. Text Classification. Continous Bag-of-words, classification with an extreme number of features (hashing of n-gram features, personalized tokens method), deep averaging network. Convolutional 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 studies: language identification, authorship attribution.

7. Sentiment analysis: classic unsupervised approaches, Ossgood's model of sentiment, feature engineering, a problem with modeling negations, sentiment lexicons, distributed representations of words, and their sentiment, sentiment analysis in Twitter.

8. Transfer learning in NLP: word embedding mapping methods (supervised and unsupervised), transfer learning from language models to text classification and other tasks, transformer architecture - BERT, Universal Sentence Encoder, GPT-3, and similar models.

9. Review of selected issues in human language technologies (selection according to students' interests): text-to-speech, speech recognition techniques (ASR), building knowledge graphs from texts, question answering, information retrieval (DSSM models), dialogue systems, topic modeling.

The above list of topics includes both lectures and tutorials - both forms of classes are an integral part of the course, i.e. the issues discussed during tutorials are often not reworked during lectures. The distribution of material between lectures and exercises is dynamic, depending on the pace of the group's work.

Teaching methods

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

2. Tutorials: presentation, illustrated with examples solved on the blackboard, practical exercises (including calculation on the blackboard), discussion of issues and solutions

Bibliography

Basic

1. Jurafsky D., Martin J.H.: Speech and Language Processing, III edycja, Pearson/Prentice Hall, 2018 (dostęp 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. Goodfellow I., Yoshua B., Courville A.: Deep Learning. MIT Press, 2016

2. 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

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

Breakdown of average student's workload

	Hours	ECTS
Total workload	120	4
Classes requiring direct contact with the teacher	60	2
Student's own work (literature studies, preparation for tutorials, preparation for tests, homeworks) [1]	60	2

1 delete or add other activities as appropriate