NLP Syllabus

1. Course Description
    a. Title of a Course: Natural Language Processing
    b. Pre-requisites: Python programming skills, general knowledge of linguistics
    c. Course Type: elective
    d. Abstract:

    The course is aimed at mastering the basics of natural language processing (NLP), a vibrant interdisciplinary field. The course covers the methods and approaches used in many real-world NLP applications such as language modeling, text classification, sentiment analysis, summarization and machine translation. The students taking the course will not only use some of the existing NLP libraries and software packages, but also learn about the principles behind their design, and about the mathematical models underlying modern computational linguistics. The course also involves completing practical programming assignments in Python and conducting experiments on texts written in English and Russian.
2. Learning Objectives:

    As a result of mastering the discipline, the student will:

    - Know the structural features of natural language texts and the principles of their computer processing in order to obtain linguistic (morphological, syntactic, semantic) information;
    - Have an idea of the methods used to solve complex practical problems of natural language processing, in particular, information retrieval, summarization, sentiment analysis, machine translation;
    - Understand the limitations of existing computer models of natural language processing.
3. Learning Outcomes

    The student will:

    - Be able to apply the existing NLP systems, determine the advantages and disadvantages of these systems, evaluate and compare the results of their work.
    - Have the skills (experience) of solving specific NLP tasks, which may involve programming in Python, as well as running experiments on textual data.

4. Course Plan

1. Introduction to natural language processing

Structural features of texts in natural language; ambiguity on all levels of language; the main challenges of natural language processing; basic approaches to problem solving: manually written rules and machine learning.

2. Basic text processing and edit distance

Preprocessing: tokenization and segmentation; normalization of words: stemming, lemmatization, morphological analyzers; regular expressions; edit distance.

3. Language models

N-grams; perplexity; methods of smoothing; the use of language models: input prediction, error correction, speech recognition, text generation.

4. Tagging problems and hidden Markov models

POS tagging; named entity recognition as a tagging problem; hidden Markov models, their ad-vantages and disadvantages; the Viterbi algorithm.

5. Text classification and sentiment analysis

Classification problems; naive Bayes classifier; text classification; sentiment analysis.

6. Evaluation

Performance measures: accuracy, precision, recall, F-measure; state-of-the-art.

7. Parsing

Constituency and dependency trees; context-free grammar; probabilistic approach to parsing; lex-icalized PCFGs; CKY algorithm.

8. Machine translation

Classical approaches: direct, transfer-based, interlingual; statistical machine translation; IBM model; alignment; parameter estimation in IBM models; phrase-based translation models.

9. Computational semantics

Word senses and meanings; WordNet; semantic similarity measures: thesaurus-based and distri-butional methods.

10. Text summarization

Extractive and abstractive summarization; multiple-document summarization; query-based sum-marization; supervised and unsupervised learning; evaluation of summarization systems; ROUGE.

5. Reading List
    a. Required
        1. Jurafsky, D., Martin, J. Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing. – Prentice Hall, 2008.
        2. Sarkar, D. Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data. – Apress, 2016.
    b. Optional
        3. ACL Anthology: A Digital Archive of Research Papers in Computational Linguistics. http://www.aclweb.org/anthology/
        4. Bird, S., Klein, E., Loper, E. Natural language processing with Python. – O'Reilly Media, Inc., 2009. http://www.nltk.org/book/
        5. Computational Linguistics. – MIT Press. http://www.mitpressjournals.org/loi/coli.
        6. Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue”. http://www.dialog-21.ru/en/digest/
6. Grading System

The students will do graded tests every two weeks, answering questions that are randomly sampled from all topics previously studied. Each test is graded for 0-10 points; the grade corresponds to the fraction of correct answers multiplied by ten, rounded (e.g. 0.75 -> 8). The final grade is the arithmetic mean of the test grades. If the final grade is lower than 8, the student is also required to take a final exam. In the exam, the student will speak on two of the following topics, assigned at random:

1. Natural Language Processing as a field today.

2. Natural language as the object of automatic processing.
3. Popular NLP tasks and general approaches to their solution.
4. Text preprocessing. Regular expressions.
5. Stemmers, lemmatizers, morphological analyzers.
6. Edit distance and its use in NLP tasks.
7. Language models: motivation and application. N-grams.
8. Problems with language models and their solutions. Evaluation methods for LMs.
9. Tagging problems; usefulness of automatic annotations.
10. Hidden Markov models, their pros and cons.
11. Text classification: task formulation and methods.
12. Naive Bayes classifier. Problems in text classification.
13. Sentiment analysis. Text representations. Feature extraction.
14. NLP system evaluation measures.
15. Information retrieval. Binary search. Term-document matrix.
16. Information retrieval. Phrase queries. TF-IDF weighting.
17. Semantic similarity. Non-distributional methods. WordNet.
18. Semantic similarity. Phrase queries. Cosine distance/similarity.
19. Python as a programming language and a tool for coding NLP scripts.
20. Machine learning in NLP. Overview of RapidMiner and Orange.

7. Methods of Instruction

   Lectures, class discussions, group projects, programming assignments, tests.

8. Special Equipment and Software Support (if required)

   The students are encouraged to use their own devices (laptops, tablets) in class whenever possible. Otherwise, classroom computers are used. The only required software is Python 3 (freeware).