Text Analytics

Ion Androutsopoulos, Associate Professor, AUEB (http://www.aueb.gr/users/ion)

Overview

The course is concerned with algorithms, models, and systems that can be used to process and extract information from natural language texts. Text analytics methods are used, for example, in sentiment analysis and opinion mining, information extraction from documents, search engines and question answering systems. They are particularly important in corporate information systems, where knowledge is often expressed in natural language (e.g., minutes, reports, regulations, contracts, product descriptions, manuals, patents). Companies also interact with their customers mostly in natural language (e.g., via e-mail, call centers, web pages describing products, blogs and social media).

Key Outcomes

By the end of the course, the students will be able to design and implement applied text analytics systems. They will also have the necessary background to pursue a research path in natural language processing.

Requirements and Prerequisites

Basic knowledge of calculus, linear algebra, probability theory. For the programming assignments, programming experience in Python is required.

Required Course Materials

There is no required textbook. Extensive notes in the form of slides are provided.

Books

There is no required textbook. Extensive notes in the form of slides are provided.

The course is mainly based on the books:

- *Speech and Language Processing*, by D. Jurafsky and J.H. Martin, 2nd edition, Pearson, 2009. The 3rd edition (in preparation) is freely available (http://web.stanford.edu/~jurafsky/slp3/).
- *Neural Network Models for Natural Language Processing*, by Y. Goldberg, Morgan & Claypool, 2017.

Both books can be found at AUEB's library.

Software/Computing Requirements

The students will be allowed to implement the programming assignments of the course in any language, though Python is recommended. An introduction to natural language processing and machine learning libraries (e.g., NLTK, scikit-learn, Keras, PyTorch) will be provided, and students will have the opportunity to use these libraries in the course's assignments. No specialized hardware is required.

Grading

In each unit, study exercises are provided (solved and unsolved, some requiring programming), of which one or two per unit are handed in (as assignments). Students are graded for their participation in class (10%), assignments (45%), and their performance at the final exam (45%).

Participation

In-class contribution is a significant part of your grade and an important part of our shared learning experience. Your active participation helps us to evaluate your overall performance. You can excel in this area if you come to class on time and contribute to the course by:

- Providing strong evidence of having thought through the material.
- Advancing the discussion by contributing insightful comments and questions.
- Listening attentively in class.
- Demonstrating interest in your peers' comments, questions, and presentations.
- Giving constructive feedback to your peers when appropriate.

Please arrive to class on time and stay to the end of the class period. Chronically arriving late or leaving class early is unprofessional and disruptive to the entire class. Repeated tardiness will have an impact on your grade.

Turn off all electronic devices prior to the start of class. Cell phones, tablets, and other electronic devices are a distraction to everyone. If the course requires you to use a laptop or other device in class, you will be informed to do so.

Late Assignments

Late assignments will either not be accepted or will incur a grade penalty unless due to documented serious illness or family emergency. Exceptions to this policy for reasons of civic obligations will only be made available when the assignment cannot reasonably be completed prior to the due date, you make suitable arrangements, and give notice for late submission in advance.

Attendance Requirements

Class attendance is essential to succeed in this course and is part of your grade. An excused absence can only be granted in cases of serious illness or grave family emergencies and must be documented. Job interviews and incompatible travel plans are considered unexcused absences. Where possible, please notify the instructor in advance of an excused absence.

Students are responsible for keeping up with the course material, including lectures, from the first day of this class, forward. It is the student's obligation to bring oneself up to date on any missed coursework.

Code of Ethics

Students may not work together on individual graded assignments unless the instructor gives express permission.

Exercise integrity in all aspects of one's academic work including, but not limited to, the preparation and completion of all other course requirements by not engaging in any method or means that provides an unfair advantage. In any case of doubt, students must be able to prove that they are the sole authors of their work by demonstrating their knowledge to the instructor.

Clearly acknowledge the work and efforts of others when submitting written work as one's own. Ideas, data, direct quotations (which should be designated with quotation marks), paraphrasing, creative expression, or any other incorporation of the work of others should be fully referenced. No plagiarism of any sort will be tolerated. This includes any material found on the internet. Reuse of material found in question and answer forums, code repositories, other lecture sites, etc., is unacceptable. You may use online material to deepen your understanding of a concept, not for finding answers.

Please report observed violations of this policy. Any violations will incur a fail grade at the course and reporting to the senate for further disciplinary action.

Course Syllabus

The course comprises ten units of three hours each.

Units 1: Introduction, n-gram language models, spelling correction, text normalization

Introduction, course organization, examples of text analytics applications. *n*-gram language models. Estimating probabilities from corpora. Entropy, cross-entropy, perplexity. Edit distance. Applications in context-sensitive spelling correction and text normalization.

Units 2 & 3: Text classification with (mostly) linear classifiers

Representing texts as bags of words. Boolean and TF-IDF features. Feature selection and extraction using information gain and SVD. Text classification with *k* nearest neighbors and Naive Bayes. Obtaining word embeddings from PMI scores. Word and text clustering with *k*-means. Linear and logistic regression, stochastic gradient descent. Lexicon-based features. Constructing and using sentiment lexica. Practical advice and diagnostics for text classification with supervised machine learning.

Unit 4 & 5: Text classification with Multi-Layer Perceptrons

Natural and artificial neural networks. Perceptrons, training them with SGD, limitations. Multi-Layer Perceptrons (MLPs) and backpropagation. Dropout. MLPs for text classification, regression, window-based sequence labelling (e.g., for POS tagging, named entity recognition). Pre-training word embeddings with Word2Vec or FastText.

Unit 6 & 7: Natural language processing with Recurrent Neural Networks

Recurrent neural networks (RNNs), GRUs/LSTMs. Applications in POS tagging and named entity recognition. RNN language models. RNNs with self-attention and applications in text classification. Bidirectional and stacked RNNs. Obtaining word embeddings from character-based RNNs. Variational dropout. Hierarchical RNNs for text classification and sequence labeling. Sequence-to-sequence RNN models with attention, and applications in machine translation. Universal sentence encoders, LASER. Pretraining language models, context-aware embeddings, ELMo.

Unit 8 & 9: Natural language processing with Convolutional Neural Networks and Transformers

Convolutional neural networks (CNNs) and applications in NLP. Image to text generation with CNN encoders and RNN decoders. Key-value attention, multi-head attention, Transformers, BERT.

Unit 10: Parsing and relation extraction

Grammars, phrase structure trees, dependency trees. Transition-based and graph-based dependency parsing with deep learning. Relation extraction as dependency parsing or via graph convolutions on parse trees.