Harvard Extension School
HARVARD DIVISION OF CONTINUING EDUCATION

CSCI E-89b
Introduction to Natural Language Processing

Fall Term 2024

Course Information

CRN: 17133
Section Number: 1
Format: Flexible Attendance Web Conference
Credit Status: Undergraduate, Graduate, Noncredit
Credit Hours: 4

Class Meetings: Mondays, September 9-December 21, 8:10pm-10:10pm

Course Description: Students are introduced to modern techniques of natural language processing (NLP) and learn foundations of text classification, named entity recognition, parsing, language modeling including text generation, topic modeling, and machine translation. Methods for representing text as data studied in the course are tokenization, n-grams, bag of words, term frequency-inverse document frequency (TD-IDF) weighting, word embeddings like Word2Vec and GloVe, autoencoders, t-SNE, character embeddings, and topic modeling. The machine learning algorithms for NLP covered in the course are recurrent neural networks (RNNs) including long short-term memory (LSTM), conditional random fields (CRFs), bidirectional LSTM with a CRF (BiLSTM-CRF), generative adversarial networks (GANs), attention models, transformers, bidirectional encoder representations from transformers (BERT), latent Dirichlet allocation (LDA), non-negative matrix factorization (NMF), and structural topic modeling (STM). Students get hands-on experience using both Python and R.

Prerequisites: Students are expected to have taken Python programming course equivalent to CSCI E-7. Most of the problems will be solved in Python. The structural topic modeling will be performed using the 'stm' R package. Prior programming

experience in R is helpful, but not required. In addition, basic knowledge of calculus, probability, and statistics is expected. Students need to have access to a computer with a 64-bit operating system and at least 8 GB of RAM. GPU is highly recommended.

Instructor Information & Office Hours

Dr. Dmitry Kurochkin
Email: dkurochkin@fas.harvard.edu
Office Hours:
By request via zoom

Section Meetings

The optional section meetings will be held on Fridays, 9:10pm-10:10pm, via zoom. Additional TA sections will be arranged. The sections will be recorded and made available in Canvas for on-demand viewing.

Course Goals / Learning Outcomes

Students will learn:

- Text Representation: Represent textual data using different methods, including tokenization, n-grams, bag of words, TF-IDF weighting, word embeddings (Word2Vec and GloVe), autoencoders, t-SNE, character embeddings, and topic modeling.
- Machine Learning Algorithms: Implement and differentiate between various machine learning algorithms used in NLP, such as RNNs, LSTMs, CRFs, BiLSTM-CRF, GANs, attention models, transformers, BERT, LDA, NMF, and STM.
- NLP Techniques and Applications: Apply NLP techniques like text classification, named entity recognition, parsing, language modeling, and machine translation in practical scenarios.

Students will gain proficiency in using Python for NLP tasks and perform structural topic modeling with the 'stm' R package.

Mode of Attendance & Participation Policy

This is a flexible attendance course, which means you can choose to: (1) attend class live over Zoom (synchronous option); or (2) watch the class recording afterward (asynchronous option). You do not need to commit to the same mode of attendance for the whole semester.

If you are attending live over Zoom:

- **Select "Zoom" in the Canvas course website to join class meetings**

Please arrive on time. You should attend Zoom meetings with a functional web-camera and microphone, prepared with materials needed, to engage thoughtfully, and with your camera on. You may turn off your camera for occasional interruptions or momentarily for privacy.

You will also need the most up-to-date Zoom client installed on your computer to join class. Please participate from a safe and appropriate environment with appropriate clothing for class. Participating while traveling or in a car is not permitted. In addition, please do not join class via mobile phone or web browser.

If you are participating asynchronously:

You are expected to watch the class recording, available in Canvas, and complete any assignments before the next live class meets.

Please be sure to review important information on **Student Policies and Conduct**.

Assignments & Grading

Assignments:

Except when especially noted, homework assignments will be due each Sunday.

Note on the deadline and penalty:

Solutions to the assignments submitted later than 1, 2, 3, 4, and 5 days after the due date will be penalized by 10%, 20%, 30%, 40%, and 100%, respectively. In case you need an extension, please coordinate with the instructor prior to the due day.

Quizzes:

An online quiz will be due before each class, unless announced otherwise. The quiz will consist of approximately 5 basic questions on understanding of studied principals. No late quizzes will be allowed.

Final Project:

The final project will be due at 11:59 pm (Eastern Time) on Monday, December 16. Late final projects will not be accepted.

Grading:

The semester average is calculated using the formula:

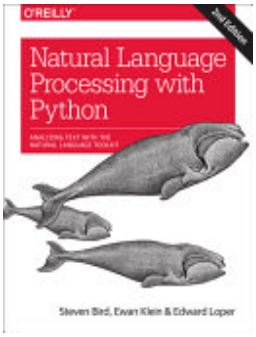
$$\text{Grade} = 0.65 \cdot \text{Homework} + 0.20 \cdot \text{Quizzes} + 0.15 \cdot \text{Final Project}$$

See Grades & Grading System for additional information.

Graduate Credit Requirements

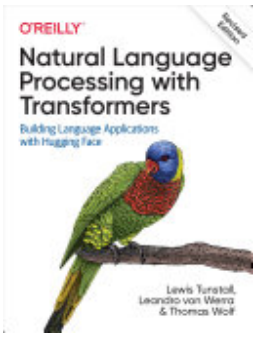
Both undergraduate and graduate students will have the same requirements and grading criteria for assignments and quizzes. The final project for graduate students will be judged at a standard of graduate-level work.

Course Materials



Natural Language Processing with Python
ISBN: 9781491913420
Authors: Steven Bird
Publication Date: 2016-01-01
Electronic copy can be obtained via Harvard Library.

Natural Language Processing with Transformers, Revised



Edition

ISBN: 9781098136765

Authors: Lewis Tunstall, Leandro von Werra, Thomas Wolf

Since their introduction in 2017, transformers have quickly become the dominant architecture for achieving state-of-the-art results on a variety of natural language processing tasks. If you're a data scientist or coder, this practical book -now revised in full color- shows you how to train and scale these large models using Hugging Face Transformers, a Python-based deep learning library. Transformers have been used to write realistic news stories, improve Google Search queries, and even create chatbots that tell corny jokes. In this guide, authors Lewis Tunstall, Leandro von Werra, and Thomas Wolf, among the creators of Hugging Face Transformers, use a hands-on approach to teach you how transformers work and how to integrate them in your applications. You'll quickly learn a variety of tasks they can help you solve. Build, debug, and optimize transformer models for core NLP tasks, such as text classification, named entity recognition, and question answering Learn how transformers can be used for cross-lingual transfer learning Apply transformers in real-world scenarios where labeled data is scarce Make transformer models efficient for deployment using techniques such as distillation, pruning, and quantization Train transformers from scratch and learn how to scale to multiple GPUs and distributed environments

Publisher: "O'Reilly Media, Inc."

Publication Date: 2022-05-26

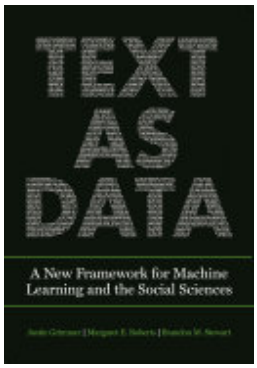
Electronic copy can be obtained via Harvard Library.

Text as Data

ISBN: 9780691207551

Authors: Justin Grimmer, Margaret E. Roberts, Brandon M. Stewart

A guide for using computational text analysis to learn about the social world From social media posts and text messages to



digital government documents and archives, researchers are bombarded with a deluge of text reflecting the social world. This textual data gives unprecedented insights into fundamental questions in the social sciences, humanities, and industry. Meanwhile new machine learning tools are rapidly transforming the way science and business are conducted. Text as Data shows how to combine new sources of data, machine learning tools, and social science research design to develop and evaluate new insights. Text as Data is organized around the core tasks in research projects using text—representation, discovery, measurement, prediction, and causal inference. The authors offer a sequential, iterative, and inductive approach to research design. Each research task is presented complete with real-world applications, example methods, and a distinct style of task-focused research. Bridging many divides—computer science and social science, the qualitative and the quantitative, and industry and academia—Text as Data is an ideal resource for anyone wanting to analyze large collections of text in an era when data is abundant and computation is cheap, but the enduring challenges of social science remain. Overview of how to use text as data Research design for a world of data deluge Examples from across the social sciences and industry
Publisher: Princeton University Press
Publication Date: 2022-03-29
Chapter 13 (Topic Models) only. Alternatively, Roberts, M., Stewart, B., & Tingley, D. (2019). stm: R Package for Structural Topic Models. *Journal of Statistical Software*, 91(2), 1-40 can be used.

Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition
ISBN: 9791221476842
Authors: Daniel Jurafsky, James H. Martin
Publication Date: 2023-01-01

This book is optional. Electronic copy of the book is available via the author's webpage: Speech and Language Processing

Academic Integrity Policy

You are responsible for understanding Harvard Extension School policies on Academic Integrity and how to use sources responsibly. Violations of academic integrity are taken very seriously. Visit Using Sources Effectively and Responsibly and the Harvard Guide to Using Sources to review important information on academic citation rules.

AI Technologies. The Extension School's Academic Integrity Policy. prohibits students from representing work as their own that they did not write, code, or create. It is never permissible to submit work generated by machine learning and AI technologies (such as ChatGPT) without proper attribution. Your instructor has the authority to set the rules governing the use of AI technology in their course, including completely prohibiting its use.

Writing Code. While it may be common practice in non-academic settings to adapt code examples found online or in texts, this is not the case in academia. In particular, you should never copy code produced as coursework by other students, whether in the current term or a previous term; nor may you provide work for other students to use. Copying code from another student or any other source is a form of academic dishonesty, as is deriving a program substantially from the work of another.

Writing code is similar to academic writing in that when you use or adapt code developed by someone else as part of your assigned coursework, you must cite your source. Paraphrasing without proper citation is just as dishonest with programming as it is with prose. A program can be considered plagiarized even though no single line is identical to any line of the source.

Accessibility Services Policy

The Division of Continuing Education (DCE) is committed to providing an accessible academic community. The Accessibility Services Office (ASO) is responsible for providing accommodations to students with disabilities. Students must request accommodations or adjustments through the ASO. Instructors cannot grant accommodation requests without prior ASO approval. It is imperative to be in touch with the ASO as soon as possible to avoid delays in the provision of accommodation.

DCE takes student privacy seriously. Any medical documentation should be provided directly to the ASO if a substantial accommodation is required. If you miss class due to a short-term illness, notify your instructor and/or TA but do not include a doctor's note. Course staff will not request, accept, or review doctor's notes or other medical documentation. For more information, email accessibility@extension.harvard.edu.

Publishing or Distributing Course Materials Policy

Students may not post, publish, sell, or otherwise distribute course materials without the written permission of the course instructor. Such materials include, but are not limited to, the following: lecture notes, lecture slides, video, or audio recordings, assignments, problem sets, examinations, other students' work, and answer keys. Students who sell, post, publish, or distribute course materials without written permission, whether for the purposes of soliciting answers or otherwise, may be subject to disciplinary action, up to and including requirement to withdraw. Further, students may not make video or audio recordings of class sessions for their own use without written permission of the instructor.

Canvas Access After End of Term

The Canvas website for this course will remain available to enrolled students for a limited time after the course concludes. **You are encouraged to download coursework and materials you wish to keep *before* the term ends.** See Course Formats & Required Technology for additional information on Canvas access.

Class Meeting Schedule

Please note that there is no class on Monday, Sept. 2, due to the University holiday

September 9: Lecture 1.

Basics of neural networks, layers, activations, forward and backward propagation, cost/loss minimization, gradient descent. Hands-on examples building and training a simple neural network in Python.

September 13: Section 1.

September 16: Lecture 2. Quiz 1 is due.

Introduction to sequences and time-series data, understanding RNNs, limitations of vanilla RNNs, introduction to LSTMs (including forget, input, and output gates), comparison with GRUs. Hands-on examples building simple RNN, LSTM, and GRU models using Python.

September 20: Section 2.

September 22: Assignment 1 is due.

September 23: Lecture 3. Quiz 2 is due.

Overview of NLP and its applications, basic text processing (tokenization, stemming, and lemmatization), introduction to NLP libraries (NLTK, SpaCy). Hands-on exercises using Python focusing on tokenization, stemming, and lemmatization.

September 27: Section 3.

September 29: Assignment 2 is due.

September 30: Lecture 4. Quiz 3 is due.

Tokenization techniques, introduction to n-grams, and Bag of Words (BoW) model. Hands-on exercises using Python for tokenization, n-grams, and implementing BoW.

October 4: Section 4.

October 6: Assignment 3 is due.

October 7: Lecture 5. Quiz 4 is due.

Term Frequency-Inverse Document Frequency (TF-IDF) weighting, introduction to word embeddings (Word2Vec, GloVe). Hands-on implementation using Python, comparison of BoW, TF-IDF, and embeddings.

October 11: Section 5.

October 13: Assignment 4 is due.

October 14: NO CLASS (University holiday)

October 21: Lecture 6. Quiz 5 is due.

Character embeddings, autoencoders, and dimensionality reduction techniques like t-SNE. Hands-on exercises using Python for character embeddings and dimensionality reduction.

October 25: Section 6.

October 27: Assignment 5 is due.

October 28: Lecture 7. Quiz 6 is due.

Introduction to topic modeling, Latent Dirichlet Allocation (LDA), and non-negative matrix factorization (NMF). Hands-on practical implementation using Python and R.

November 1: Section 7.

November 3: Assignment 6 is due.

November 4: Lecture 8. Quiz 7 is due.

Introduction to structural topic modeling and the 'stm' R package. Hands-on practical implementation of structural topic modeling using R.

November 8: Section 8.

November 10: Assignment 7 is due.

November 11: Lecture 9. Quiz 8 is due.

Logistic Regression, Naive Bayes classifiers, Support Vector Machines (SVM), Decision Trees, Random Forests, and k-nearest neighbors (k-NN), comparison of classification methods. Hands-on implementation using Python.

November 15: Section 9.

November 17: Assignment 8 is due.

November 18: Lecture 10. Quiz 9 is due.

Named Entity Recognition (NER), introduction to rule-based and statistical methods. Hands-on examples using Python (NLTK, SpaCy) for NER.

November 22: Section 10.

November 24: Assignment 9 is due.

November 25: Lecture 11. Quiz 10 is due.

Conditional Random Fields (CRFs), Bidirectional LSTM with CRF (BiLSTM-CRF), and introduction to GANs. Hands-on exercises with advanced models using Python.

December 2: Lecture 12. Quiz 11 is due.

Understanding transformers and BERT, fine-tuning BERT for downstream tasks. Hands-on practical implementation using Python.

December 6: Section 11.

December 8: Assignment 10 is due.

December 9: Lecture 13. Quiz 12 is due.

Basics of machine translation, sequence-to-sequence models with attention. Hands-on practical implementation of machine translation tasks using Python.

December 13: Section 12.

December 16: Final Project.

Final Exam

Final Project is due December 16, 11:59 pm (Eastern Time).