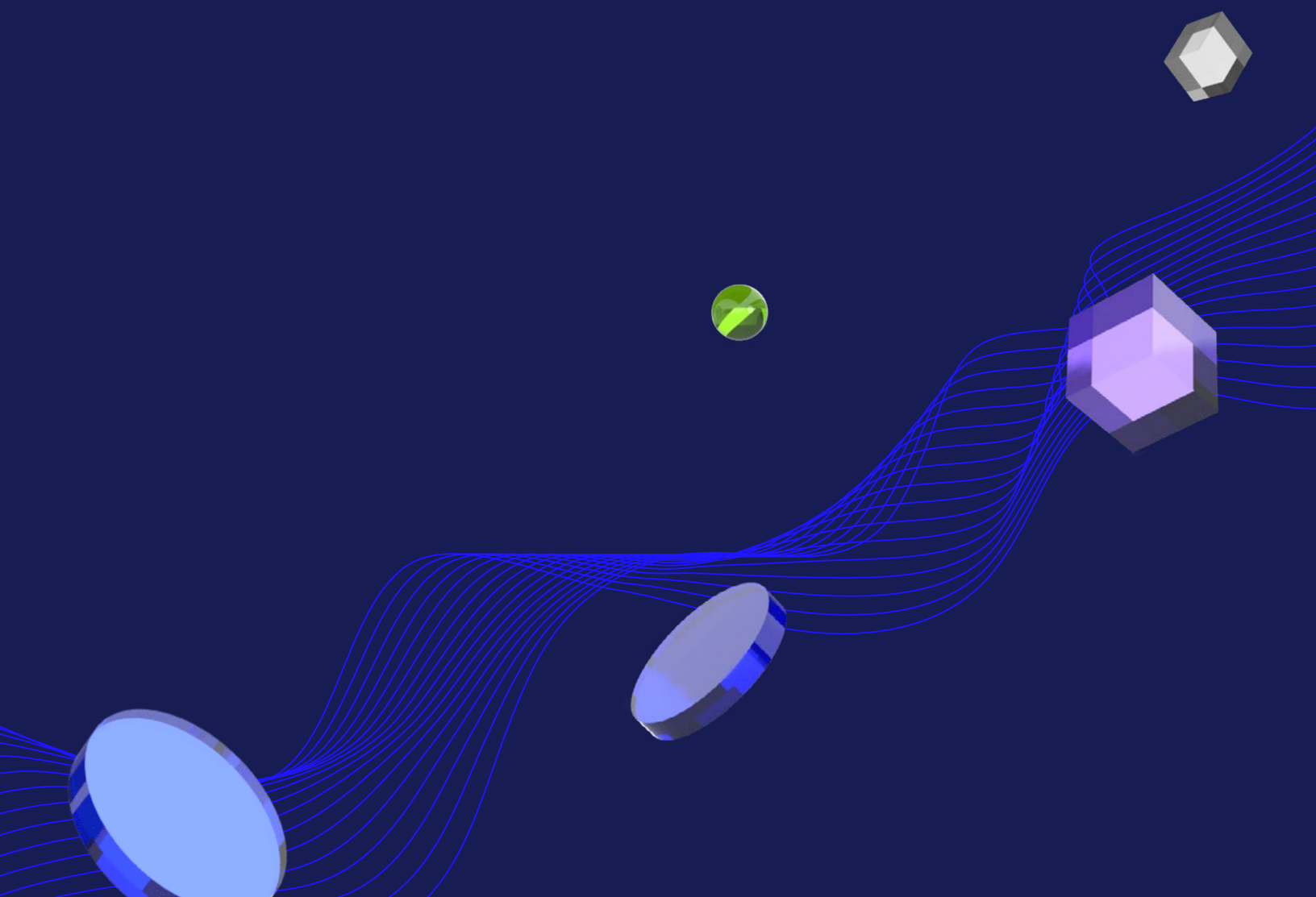
INDIVIDUAL LEARNERS

UDACITY

SCHOOL OF ARTIFICIAL INTELLIGENCE

Natural Language Processing

Nanodegree Program Syllabus



Overview

This program will enhance learners' existing machine learning and deep learning skills with the addition of natural language processing and speech recognition techniques. These skills can be used in various applications such as part of speech tagging and machine translation, among others. Learners will develop the skills they need to start applying natural language processing techniques to real-world challenges and applications.

Built in collaboration with:

Program information

Estimated Time

3 months at 10-15hrs/week*

Skill Level

Advanced

Prerequisites

A well-prepared learner should have significant experience with Python and entry-level experience with probability, statistics, and deep learning architectures. Learners should also have the ability to write a class in Python and add comments to their code for others to read. Lastly, learners should have familiarity with the term “neural networks” and the differential math that drives backpropagation.

Required Hardware/Software

Learners need access to a 64-bit operating system with at least 8GB of RAM, along with administrator account permissions sufficient to install programs including Anaconda with Python 3.5 and supporting packages. The network should allow secure connections to remote hosts (like SSH).

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

Introduction to Natural Language Processing

Learn text processing fundamentals including stemming and lemmatization. Explore machine learning methods in sentiment analysis. Build a speech tagging model.

Course Project

Part of Speech Tagging

Use several techniques, including table lookups, n-grams, and hidden Markov models, to tag parts of speech in sentences, and compare their performance. This project demonstrates text processing techniques that allow one to build a part of speech tagging model. Work with a simple lookup table and progressively add more complexity to improve the model using probabilistic graphical models. Use a Python package to build and train a tagger with a hidden Markov model, and compare the performances of all these models in a data set of sentences.

Lesson 1

Intro to NLP

- Learn the main techniques used in natural language processing.
- Get familiarized with the terminology and the topics in the class.
- Build one's first application with IBM Watson.

Lesson 2

Text Processing

- See how text gets processed in order to use it in models.
- Learn techniques such as tokenization, stemming, and lemmatization.
- Get started with part of speech tagging and named entity recognition.

Lesson 3

Part of Speech Tagging with Hidden Markov Models

- Learn how hidden Markov models are defined.
- Train HMMs with the Viterbi and the Baum-Welch algorithms.
- Use HMMs to build a part of speech tagging model.

Course 2

Computing with Natural Language

Learn advanced techniques like word embeddings, deep learning attention, and more. Build a machine translation model using recurrent neural network architectures.

Course Project

Machine Translation

Build a deep neural network that functions as part of an end-to-end machine translation pipeline. The completed pipeline will accept English text as input and return the French translation. Be able to explore several recurrent neural network architectures and compare their performance. Pre-process the data by converting text to sequence of integers. Build several deep learning models for translating the text into French. Run this models on English test to analyze their performance.

Lesson 1

Feature Extraction & Embeddings

- Learn to extract features from text.
- Learn the most used embedding algorithms, such as Word2Vec and Glove.

Lesson 2

Modeling

- Learn about the main uses of deep learning models in NLP.
- Learn about machine translation, topic models, and sentiment analysis.

Lesson 3

Deep Learning Attention

- Learn about attention, the advanced deep learning method empowering applications like Google Translate.
- Learn about additive and multiplicative attention in applications like machine translation, text summarization, and image captioning.
- Learn about cutting-edge deep learning models like the transformer that extend the use of attention to eliminate the need for RNNs.

Lesson 4

Information Systems

- Learn about information extraction and information retrieval systems.
- Learn about question answering and its applications.

Course 3

Communicating with Natural Language

Learn voice user interface techniques that turn speech into text and vice versa. Build a speech recognition model using deep neural networks.

Course Project

Speech Recognizer

Build a deep neural network that functions as part of an end-to-end automatic speech recognition (ASR pipeline. The model will convert raw audio into feature representations, which will then turn them into transcribed text. Begin by investigating a data set that will be used to train and evaluate the models. Convert any raw audio to feature representations that are commonly used for ASR. Build neural networks that map these features to transcribed text.

Lesson 1

Intro to Voice User Interfaces

- Learn the basics of how computers understand spoken words.
- Get familiar with the most common VUI applications.
- Set up an AWS account and build Alexa skill with an existing template.

Lesson 2

Alexa History SKill

- Learn the basics of Amazon AWS.
- Create one's own fully functional Alexa skill using Amazon's API.
- Deploy one's skill for everyone to use it.

Lesson 3

Introduction to Speech Recognition

- Learn the pipeline used for speech recognition.
- Learn to process and extract features from sound signals.
- Learn to build probabilistic and machine learning language models in order to extract words and grammar from sound signals.

Meet your instructors.

Luis Serrano

Instructor

Luis was formerly a machine learning engineer at Google. He holds a PhD in mathematics from the University of Michigan, and a postdoctoral fellowship at the University of Quebec at Montreal.

Jay Alammar

Investment Principal at STV

Jay has a degree in computer science, loves visualizing machine learning concepts, and is the investment principal at STV, a $500 million venture capital fund focused on high-technology startups.

Arpan Chakraborty

Computer Scientist

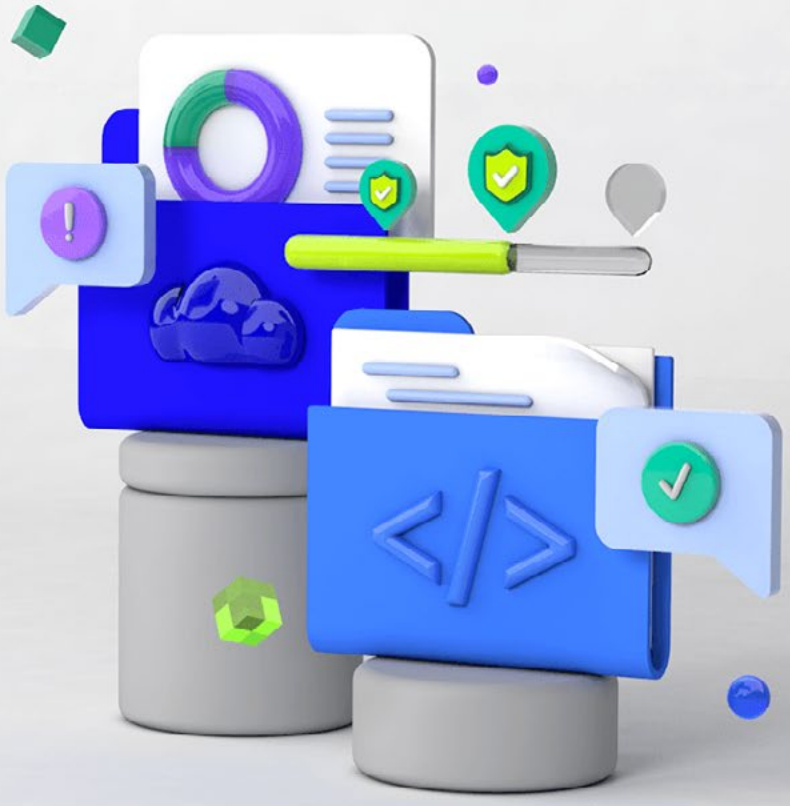
Arpan is a computer scientist with a PhD from North Carolina State University. He teaches at Georgia Tech (within the Master of Computer Science program), and is a coauthor of the book Practical Graph Mining with R.

Dana Sheahen

Electrical Engineer

Dana is an electrical engineer with a master's in computer science from Georgia Tech. Her work experience includes software development for embedded systems in the Automotive Group at Motorola, where she was awarded a patent for an onboard operating system.

Udacity's learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren't just multiple choice questions or step-by-step guides, but instead require critical thinking.

Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.

Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

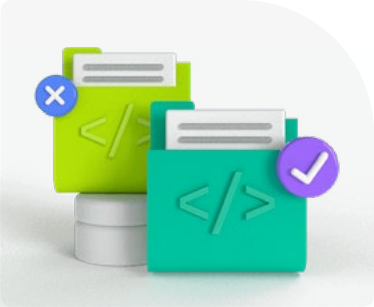
Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

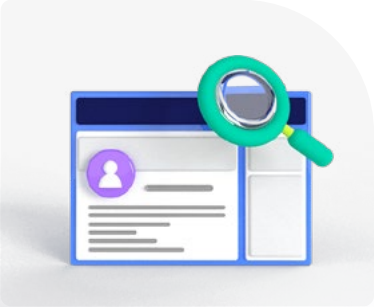
Verify skills mastery.

- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.

Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.

Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

UDACITY

Learn more at

www.udacity.com/online-learning-for-individuals →

12.22.22 | V1.0