Natural Language Processing Nanodegree Syllabus

Master the skills to get computers to understand, process and manipulate human language.

UDACITY
FOR ENTERPRISE

Natural Language Processing

Table of Contents

Natural Language Processing

Before You Start

Prerequisites: In order to succeed in this program, we recommend having significant experience with Python, and entry-level experience with probability and statistics, and deep learning architectures. Specifically, we expect you to be able to write a class in Python, and to add comments to your code for others to read. Also, you should be familiar with "neural networks," and understand the differential math that drives backpropagation.

Educational Objectives: In this program, you'll learn and put into practice the concepts behind natural language processing and speech recognition, including machine translation, part of speech tagging, and sentiment analysis. This course will cover the latest in deep learning architectures, which, combined with the more classical probabilistic methods used in a variety of applications, will give you a repertory of techniques widely used in the industry. With the practical skills you gain in this course, you'll be able to build your own models which will extract information from text or voice data, and enable you to solve real-world challenges.

Contact Info

While going through the program, if you have questions about anything, you can reach us at enterprise-support@udacity.com. For help from Udacity mentors and peers, please visit the Udacity classroom.

Nanodegree Program Info

Technical Requirements
Hardware Requirements: 64-bit operating system with 8GB of RAM, webcam, microphone
Software and Software Version Requirements: Text editor, Python 3.6, Jupyter Notebooks, Anaconda, PyTorch, Keras, OpenCV
LENGTH OF PROGRAM*: 4 months
FREQUENCY OF CLASSES: Self-paced
TEXTBOOKS REQUIRED: None

*This is a self-paced program and the length is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.

Project 1: 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 you to build a part of speech tagging model. You will work with a simple lookup table, and progressively add more complexity to improve the model using probabilistic graphical models. Ultimately you'll be using a Python package to build and train a tagger with a hidden Markov model, and you will be able to compare the performances of all these models in a dataset of sentences.

Supporting Lesson Content: Introduction to Natural Language Processing

Lesson Title	Learning Outcomes
INTRO TO NLP	• Learn the main techniques used in natural language processing. • Get familiarized.with the terminology and the topics in the class. • Build your first application with IBM Watson.
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.
PART OG 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 models.

Project 2: Machine Translation

Build a deep neural network that functions as part of an end-to-end machine translation pipeline. Your completed pipeline will accept English text as input and return the French translation. You'll be able to explore several recurrent neural network architectures and compare their performance. First you will preprocess the data by converting text to sequence of integers. Then you will build several deep learning models for translating the text into French. As a final step, you will run this models on English test to analyze their performance.

Supporting Lesson Content: Computing with Natural Language

Lesson Title	Learning Outcomes
FEATURE EXTRACTION AND EMBEDDINGS	• Express logical constraints as Python functions. • Use constraint propagation & search to solve all Sudoku puzzles.
MODELING	• Learn to represent problems in terms of logical constraints. • Use constraint propagation to limit the potential solution space. • Incorporate backtracking search to find a solution when the set of constraints is incomplete.
DEEP LEARNING ATTENTION	• List of external resources for you to continue learning about CSPs.
INFORMATION SYSTEMS	• Learn about information extraction and information retrieval systems. • Learn about question answering and its applications.

Project 3: 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. You'll begin by investigating a dataset, that will be used to train and evaluate your models. Your algorithm will first convert any raw audio to feature representations that are commonly used for ASR. You will then build neural networks that map these features to transcribed text.

Supporting Lesson Content: Communicating with Natural Language

Lesson Title	Learning Outcomes
INTRODUCTION TO VOICE USER INTERFACES	• Learn the basics of how computers understand spoken words. • Get familiar with the most common VUI applications. • Set up your AWS account and build Alexa skill with an existing template.
ALEXA HISTORY SKILL	• Learn the basics of Amazon AWS. • Create your own fully functional Alexa skill using Amazon's API. • Deploy your skill for everyone to use it.
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.