Hands-On Natural Language Processing with Python

A practical guide to applying deep learning architectures to your NLP applications

By Rajesh Arumugam
and Rajalingappaa Shanmugamani

Packt
www.packt.com

Hands-On Natural Language Processing with Python

A practical guide to applying deep learning architectures to your NLP applications

Rajesh Arumugam
Rajalingappaa Shanmugamani

Packt>

BIRMINGHAM - MUMBAI

Hands-On Natural Language Processing with Python

Commissioning Editor: Pravin Dhandre
Acquisition Editor: Aman Singh
Content Development Editor: Snehal Kolte
Technical Editor: Sayli Nikalje
Copy Editor: Safis Editing
Project Coordinator: Manthan Patel
Proofreader: Safis Editing
Indexer: Pratik Shirodkar
Graphics: Jisha Chirayil
Production Coordinator: Nilesh Mohite

First published: July 2018

Production reference: 1160718

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-78913-949-5

www.packtpub.com

Table of Contents

Preface 1

Chapter 1: Getting Started 9
- **Basic concepts and terminologies in NLP** 9
  - Text corpus or corpora 10
  - Paragraph 10
  - Sentences 10
  - Phrases and words 11
  - N-grams 11
  - Bag-of-words 11
- **Applications of NLP** 12
  - Analyzing sentiment 12
  - Recognizing named entities 13
  - Linking entities 14
  - Translating text 15
  - Natural Language Inference 17
  - Semantic Role Labeling 17
  - Relation extraction 18
  - SQL query generation, or semantic parsing 18
  - Machine Comprehension 20
  - Textual Entailment 22
  - Coreference resolution 22
  - Searching 23
  - Question answering and chatbots 24
  - Converting text-to-voice 24
  - Converting voice-to-text 26
  - Speaker identification 26
  - Spoken dialog systems 27
  - Other applications 27
- **Summary** 28

Chapter 2: Text Classification and POS Tagging Using NLTK 29
- **Installing NLTK and its modules** 29
- **Text preprocessing and exploratory analysis** 31
  - Tokenization 31
  - Stemming 33
  - Removing stop words 34
- **Exploratory analysis of text** 35
- **POS tagging** 38
  - What is POS tagging? 39

Applications of POS tagging 40
Training a POS tagger 41
Training a sentiment classifier for movie reviews 45
Training a bag-of-words classifier 50
Summary 52

Chapter 3: Deep Learning and TensorFlow 53
Deep learning 53
Perceptron 54
Activation functions 54
Sigmoid 55
Hyperbolic tangent 56
Rectified linear unit 56
Neural network 57
One-hot encoding 58
Softmax 58
Cross-entropy 59
Training neural networks 59
Backpropagation 59
Gradient descent 60
Stochastic gradient descent 61
Regularization techniques 61
Dropout 62
Batch normalization 62
L1 and L2 normalization 62
Convolutional Neural Network 63
Kernel 63
Max pooling 64
Recurrent neural network 64
Long-Short Term Memory 65
TensorFlow 65
General Purpose – Graphics Processing Unit 65
CUDA 66
cuDNN 67
Installation 67
Hello world! 67
Adding two numbers 68
TensorBoard 69
The Keras library 70
Summary 70

Chapter 4: Semantic Embedding Using Shallow Models 71
Word vectors 71
The classical approach 72
Word2vec 73
The CBOW model 75
The skip-gram model 75

A comparison of skip-gram and CBOW model architectures 75
Building a skip-gram model 76
Visualization of word embeddings 79
From word to document embeddings 84
Sentence2vec 84
Doc2vec 86
Visualization of document embeddings 87
Summary 90

Chapter 5: Text Classification Using LSTM 91
Data for text classification 92
Topic modeling 92
Topic modeling versus text classification 96
Deep learning meta architecture for text classification 96
Embedding layer 97
Deep representation 97
Fully connected part 97
Identifying spam in YouTube video comments using RNNs 98
Classifying news articles by topic using a CNN 103
Transfer learning using GloVe embeddings 108
Multi-label classification 112
Binary relevance 112
Deep learning for multi-label classification 112
Attention networks for document classification 114
Summary 116

Chapter 6: Searching and DeDuplicating Using CNNs 117
Data 118
Data description 118
Training the model 118
Encoding the text 120
Modeling with CNN 122
Training 124
Inference 127
Summary 128

Chapter 7: Named Entity Recognition Using Character LSTM 129
NER with deep learning 130
Data 130
Model 132
Word embeddings 132
Walking through the code 133
Input 134
Word embedding 134
The effects of different pretrained word embeddings 136
Neural network architecture 138

Decoding predictions 140
The training step 141
Scope for improvement 147
Summary 148

Chapter 8: Text Generation and Summarization Using GRUs 149
Generating text using RNNs 149
Generating Linux kernel code with a GRU 150
Text summarization 156
Extractive summarization 157
Summarization using gensim 157
Abstractive summarization 159
Encoder-decoder architecture 159
Encoder 159
Decoder 159
News summarization using GRU 160
Data preparation 160
Encoder network 163
Decoder network 163
Sequence to sequence 165
Building the graph 165
Training 166
Inference 168
TensorBoard visualization 169
State-of-the-art abstractive text summarization 170
Summary 173

Chapter 9: Question-Answering and Chatbots Using Memory Networks 175
The Question-Answering task 175
Question-Answering datasets 176
Memory networks for Question-Answering 176
Memory network pipeline overview 177
Writing a memory network in TensorFlow 178
Class constructor 178
Input module 179
Question module 180
Memory module 180
Output module 182
Putting it together 183
Extending memory networks for dialog modeling 184
Dialog datasets 185
The bAbI dialog dataset 185
Raw data format 187
Writing a chatbot in TensorFlow 188
Loading dialog datasets in the QA format 188
Vectorizing the data 190
Wrapping the memory network model in a chatbot class 192
Class constructor 192

Building a vocabulary for word embedding lookup 193
Training the chatbot model 193
Evaluating the chatbot on the testing set 195
Interacting with the chatbot 195
Putting it all together 196
Example of an interactive conversation 197
Literature on and related to memory networks 198
Summary 199

Chapter 10: Machine Translation Using the Attention-Based Model 201
Overview of machine translation 201
Statistical machine translation 202
English to French using NLTK SMT models 203
Neural machine translation 205
Encoder-decoder network 205
Encoder-decoder with attention 207
NMT for French to English using attention 208
Data preparation 208
Encoder network 211
Decoder network 212
Sequence-to-sequence model 213
Building the graph 214
Training 215
Inference 217
TensorBoard visualization 220
Summary 221

Chapter 11: Speech Recognition Using DeepSpeech 223
Overview of speech recognition 223
Building an RNN model for speech recognition 224
Audio signal representation 224
LSTM model for spoken digit recognition 226
TensorBoard visualization 228
Speech to text using the DeepSpeech architecture 230
Overview of the DeepSpeech model 230
Speech recordings dataset 231
Preprocessing the audio data 232
Creating the model 234
TensorBoard visualization 239
State-of-the-art in speech recognition 240
Summary 242

Chapter 12: Text-to-Speech Using Tacotron 243
Overview of text to speech 244
Naturalness versus intelligibility 244
How is the performance of a TTS system evaluated? 245
Traditional techniques – concatenative and parametric models 245
A few reminders on spectrograms and the mel scale 246
TTS in deep learning 250

WaveNet, in brief 250
Tacotron 251
The encoder 252
The attention-based decoder 255
The Griffin-Lim-based postprocessing module 256
Details of the architecture 256
Limitations 256
Implementation of Tacotron with Keras 257
The dataset 258
Data preparation 259
Preparation of text data 260
Preparation of audio data 261
Implementation of the architecture 264
Pre-net 264
Encoder and postprocessing CBHG 264
Attention RNN 266
Decoder RNN 266
The attention mechanism 267
Full architecture, with attention 267
Training and testing 269
Summary 270

Chapter 13: Deploying Trained Models 271
Increasing performance 271
Quantizing the weights 272
MobileNets 272
TensorFlow Serving 275
Exporting the trained model 277
Serving the exported model 277
Deploying in the cloud 278
Amazon Web Services 278
Google Cloud Platform 282
Deploying on mobile devices 285
iPhone 285
Android 285
Summary 286

Other Books You May Enjoy 287

Index 291

Preface

Before the advent of deep learning, traditional **natural language processing** (**NLP**) approaches had been widely used in tasks such as spam filtering, sentiment classification, and **part of speech** (**POS**) tagging. These classic approaches utilized statistical characteristics of sequences such as word count and co-occurrence, as well as simple linguistic features. However, the main disadvantage of these techniques was that they could not capture complex linguistic characteristics, such as context and intra-word dependencies.

Recent developments in neural networks and deep learning have given us powerful new tools to match human-level performance on NLP tasks and build products that deal with natural language. Deep learning for NLP is centered around the concept of word embeddings or vectors, also known as Word2vec, which encapsulate the meanings of words and phrases as dense vector representations. Word vectors, which are able to capture semantic information about words better than traditional one-hot representations, allow us to handle the temporal nature of language in an intuitive way when used in combination with a class of neural networks known as **recurrent neural networks** (**RNNs**). While RNNs can capture only local word dependencies, recently proposed vector-based operations for attention and alignment over word vector sequences allow neural networks to model global intra-word dependencies, including context. Due to their capability to model the syntax and semantics of language, strong empirical performance, and ability to generalize to new data, neural networks have become the go-to model for building highly sophisticated commercial products, such as search engines, translation services, and dialog systems.

This book introduces the basic building blocks of deep learning models for NLP and explores cutting-edge techniques from recent literature. We take a problem-based approach, where we introduce new models as solutions to various NLP tasks. Our focus is on providing practical code implementations in Python that can be applied to your use cases to bring human capabilities into your applications.

Who this book is for

This book is intended for developers who want to leverage NLP techniques to develop intelligent applications with rich human-centric interfaces. The book assumes introductory knowledge of **machine learning** (**ML**) or deep learning and intermediate Python programming skills. Our aim is to introduce cutting-edge techniques for NLP tasks, such as sentiment detection, conversational systems, language translation, speech-to-text, and much more, using the TensorFlow framework and Python.

The reader will go from the basic concepts of deep learning to state-of-the-art algorithms and best practices for dealing with natural language. Our focus is on implementing applications using real-world data and deploying deep learning models to add human capabilities to commercial applications in a production environment.

What this book covers

`Chapter 1`, *Getting Started*, explores the basic concepts of NLP and the various problems it tries to solve. We also look at some of the real-world applications to give the reader the feeling of the wide range of applications that leverage NLP.

`Chapter 2`, *Text Classification and POS Tagging Using NLTK*, introduces the popular NLTK Python library. We will be using NLTK to describe basic NLP tasks, such as tokenizing, stemming, tagging, and classic text classification. We also explore POS tagging with NLTK. We provide the reader with the tools and techniques necessary to prepare data for input into deep learning models.

`Chapter 3`, *Deep Learning and TensorFlow*, introduces the basic concepts of deep learning. This chapter will also help the reader to set up the environment and tools such as TensorFlow. At the end of the chapter, the reader will get an understanding of basic deep learning concepts, such as CNN, RNN, LSTM, attention-based models, and problems in NLP.

`Chapter 4`, *Semantic Embedding Using Shallow Models*, explores how to identify semantic relationships between words in a document, and in the process, we obtain a vector representation for words in a corpus. The chapter describes developing word embedding models, such as CBOW using neural networks. It also describes techniques for developing neural network models to obtain document vectors. At the end of this chapter, the reader will get familiar with training embeddings for word, sentence, and document; and visualize simple networks.

`Chapter 5`, *Text Classification Using LSTM*, discusses various approaches for classifying text, a specific application of which is to classify sentiments of words or phrases in a document. The chapter introduces the problem of text classification. Following this, we describe techniques for developing deep learning models using CNNs and LSTMs. The chapter also explains transfer learning for text classification using pretrained word embeddings. At the end, the reader will get familiar with implementing deep learning models for sentiment classification, spam detection, and using pretrained word embeddings for his/her classification task.

`Chapter 6`, *Searching and Deduplicating Using CNNs*, covers the problems of searching, matching and deduplicating documents and approaches used in solving them. The chapter describes developing deep learning models for searching text in a corpus. At the end of this chapter, you will learn to implement a CNN-based deep learning model for searching and deduplicating text.

`Chapter 7`, *Named Entity Recognition Using Character LSTM*, describes methods and approaches to perform **Named Entity Recognition** (**NER**), a sub-task of information extraction, to locate and classify entities in text of a document. The chapter introduces the problem of NER and the applications where it can be used. We then explain the implementation of a deep learning model using character-based LSTM for identifying named entities trained using labeled datasets.

`Chapter 8`, *Text Generation and Summarization Using GRUs*, covers the methods used for the task of generating text, an extension of which can be used to create summaries from text data. We then explain the implementation of a deep learning model for generating text. This is followed by a description of implementing GRU-based deep learning models to summarize text. At the end of this chapter, the reader will learn the techniques of implementing deep learning models for text generation and summarization.

`Chapter 9`, *Question-Answering and Chatbots Using Memory Networks*, describes how to train a deep learning model to answer questions and extend it to build a chatbot. The chapter introduces the problem of question answering and the approaches used in building an answering engine using deep learning models. We then describe how to leverage a question-answering engine to build a chatbot capable of answering questions like a conversation. At the end of this chapter, you will be able to implement an interactive chatbot.

`Chapter 10`, *Machine Translation Using Attention-Based Models*, covers various methods for translating text from one language to another, without the need to learn the grammar structure of either language. The chapter introduces traditional machine translation approaches, such as **Hidden Markov Model** (**HMM**) based methods. We then explain the implementation of an encoder-decoder model with attention for translating text from French to the English language. At the end of this chapter, the reader will be able to implement deep learning models for translating text.

`Chapter 11`, *Speech Recognition Using Deep Speech*, describes the problem of converting voice to text, as a beginning of a conversational interface. The chapter begins with feature extraction from speech data. This is followed by a brief introduction of the deep speech architecture. We then explain the detailed implementation of the Deep Speech architecture to transcribe speech to text. At the end of this chapter, the reader will be equipped with the knowledge to implement a speech-to-text deep learning model.

`Chapter 12`, *Text to Speech Using Tacotron*, describes the problem of converting text to speech. The chapter describes the implementation of the Tacotron model to convert text to voice. At the end, the reader will get familiar with the implementation of a text-to-speech model based on the Tacotron architecture.

`Chapter 13`, *Deploying Trained Models*, is the concluding chapter and describes model deployments in various cloud and mobile platforms.

To get the most out of this book

The prerequisites for the book are basic knowledge of ML or deep learning and intermediate Python skills, although both are not mandatory. We have given a brief introduction to deep learning, touching upon topics such as multi-layer perceptrons, **Convolutional Neural Networks** (**CNNs**), and RNNs in `Chapter 1`, *Getting Started*. It would be helpful if the reader knows general ML concepts, such as overfitting and model regularization, and classical models, such as linear regression and random forest. In more advanced chapters, the reader might encounter in-depth code walkthroughs that expect at least a basic level of Python programming experience.