

Table of Contents

Preface	xi
Chapter 1: Introduction to Natural Language Processing	1
What is Natural Language Processing?	1
Tasks of Natural Language Processing	2
The traditional approach to Natural Language Processing	5
Understanding the traditional approach	5
Example – generating football game summaries	6
Drawbacks of the traditional approach	10
The deep learning approach to Natural Language Processing	10
History of deep learning	11
The current state of deep learning and NLP	13
Understanding a simple deep model – a Fully-Connected Neural Network	14
The roadmap – beyond this chapter	16
Introduction to the technical tools	21
Description of the tools	21
Installing Python and scikit-learn	22
Installing Jupyter Notebook	22
Installing TensorFlow	23
Summary	24
Chapter 2: Understanding TensorFlow	27
What is TensorFlow?	28
Getting started with TensorFlow	28
TensorFlow client in detail	31
TensorFlow architecture – what happens when you execute the client?	32
Cafe Le TensorFlow – understanding TensorFlow with an analogy	35
Inputs, variables, outputs, and operations	36
Defining inputs in TensorFlow	37

Feeding data with Python code	37
Preloading and storing data as tensors	38
Building an input pipeline	39
Defining variables in TensorFlow	43
Defining TensorFlow outputs	45
Defining TensorFlow operations	45
Comparison operations	45
Mathematical operations	46
Scatter and gather operations	47
Neural network-related operations	48
Reusing variables with scoping	57
Implementing our first neural network	59
Preparing the data	60
Defining the TensorFlow graph	61
Running the neural network	63
Summary	65
Chapter 3: Word2vec – Learning Word Embeddings	67
<hr/>	
What is a word representation or meaning?	69
Classical approaches to learning word representation	69
WordNet – using an external lexical knowledge base for learning word representations	70
Tour of WordNet	70
Problems with WordNet	73
One-hot encoded representation	74
The TF-IDF method	75
Co-occurrence matrix	76
Word2vec – a neural network-based approach to learning word representation	77
Exercise: is queen = king – he + she?	78
Designing a loss function for learning word embeddings	82
The skip-gram algorithm	83
From raw text to structured data	83
Learning the word embeddings with a neural network	84
Formulating a practical loss function	87
Efficiently approximating the loss function	90
Implementing skip-gram with TensorFlow	95
The Continuous Bag-of-Words algorithm	98
Implementing CBOW in TensorFlow	99
Summary	100
Chapter 4: Advanced Word2vec	103
<hr/>	
The original skip-gram algorithm	104
Implementing the original skip-gram algorithm	105

Comparing the original skip-gram with the improved skip-gram	107
Comparing skip-gram with CBOW	107
Performance comparison	108
Which is the winner, skip-gram or CBOW?	112
Extensions to the word embeddings algorithms	114
Using the unigram distribution for negative sampling	114
Implementing unigram-based negative sampling	115
Subsampling – probabilistically ignoring the common words	117
Implementing subsampling	118
Comparing the CBOW and its extensions	118
More recent algorithms extending skip-gram and CBOW	119
A limitation of the skip-gram algorithm	119
The structured skip-gram algorithm	120
The loss function	120
The continuous window model	122
GloVe – Global Vectors representation	123
Understanding GloVe	123
Implementing GloVe	125
Document classification with Word2vec	126
Dataset	127
Classifying documents with word embeddings	127
Implementation – learning word embeddings	128
Implementation – word embeddings to document embeddings	129
Document clustering and t-SNE visualization of embedded documents	130
Inspecting several outliers	131
Implementation – clustering/classification of documents with K-means	132
Summary	134
Chapter 5: Sentence Classification with Convolutional Neural Networks	135
Introducing Convolution Neural Networks	136
CNN fundamentals	136
The power of Convolution Neural Networks	139
Understanding Convolution Neural Networks	139
Convolution operation	140
Standard convolution operation	140
Convolving with stride	141
Convolving with padding	142
Transposed convolution	143
Pooling operation	144
Max pooling	145
Max pooling with stride	145
Average pooling	146

Fully connected layers	147
Putting everything together	147
Exercise – image classification on MNIST with CNN	148
About the data	149
Implementing the CNN	149
Analyzing the predictions produced with a CNN	152
Using CNNs for sentence classification	153
CNN structure	153
Data transformation	153
The convolution operation	154
Pooling over time	157
Implementation – sentence classification with CNNs	159
Summary	162
Chapter 6: Recurrent Neural Networks	163
Understanding Recurrent Neural Networks	164
The problem with feed-forward neural networks	165
Modeling with Recurrent Neural Networks	166
Technical description of a Recurrent Neural Network	168
Backpropagation Through Time	170
How backpropagation works	170
Why we cannot use BP directly for RNNs	171
Backpropagation Through Time – training RNNs	172
Truncated BPTT – training RNNs efficiently	173
Limitations of BPTT – vanishing and exploding gradients	173
Applications of RNNs	175
One-to-one RNNs	176
One-to-many RNNs	176
Many-to-one RNNs	177
Many-to-many RNNs	178
Generating text with RNNs	179
Defining hyperparameters	179
Unrolling the inputs over time for Truncated BPTT	180
Defining the validation dataset	181
Defining weights and biases	181
Defining state persisting variables	181
Calculating the hidden states and outputs with unrolled inputs	182
Calculating the loss	183
Resetting state at the beginning of a new segment of text	183
Calculating validation output	184
Calculating gradients and optimizing	184
Outputting a freshly generated chunk of text	184

Evaluating text results output from the RNN	185
Perplexity – measuring the quality of the text result	187
Recurrent Neural Networks with Context Features – RNNs with longer memory	188
Technical description of the RNN-CF	188
Implementing the RNN-CF	190
Defining the RNN-CF hyperparameters	190
Defining input and output placeholders	191
Defining weights of the RNN-CF	191
Variables and operations for maintaining hidden and context states	192
Calculating output	194
Calculating the loss	195
Calculating validation output	195
Computing test output	196
Computing the gradients and optimizing	196
Text generated with the RNN-CF	196
Summary	199
Chapter 7: Long Short-Term Memory Networks	201
Understanding Long Short-Term Memory Networks	202
What is an LSTM?	203
LSTMs in more detail	204
How LSTMs differ from standard RNNs	212
How LSTMs solve the vanishing gradient problem	213
Improving LSTMs	216
Greedy sampling	217
Beam search	218
Using word vectors	219
Bidirectional LSTMs (BiLSTM)	220
Other variants of LSTMs	222
Peephole connections	223
Gated Recurrent Units	224
Summary	226
Chapter 8: Applications of LSTM – Generating Text	229
Our data	230
About the dataset	230
Preprocessing data	232
Implementing an LSTM	232
Defining hyperparameters	232
Defining parameters	233
Defining an LSTM cell and its operations	235
Defining inputs and labels	236
Defining sequential calculations required to process sequential data	237

Defining the optimizer	238
Decaying learning rate over time	238
Making predictions	240
Calculating perplexity (loss)	240
Resetting states	240
Greedy sampling to break unimodality	241
Generating new text	241
Example generated text	242
Comparing LSTMs to LSTMs with peephole connections and GRUs	243
Standard LSTM	243
Review	243
Example generated text	244
Gated Recurrent Units (GRUs)	245
Review	245
The code	246
Example generated text	247
LSTMs with peepholes	248
Review	248
The code	248
Example generated text	249
Training and validation perplexities over time	250
Improving LSTMs – beam search	251
Implementing beam search	252
Examples generated with beam search	254
Improving LSTMs – generating text with words instead of n-grams	255
The curse of dimensionality	255
Word2vec to the rescue	255
Generating text with Word2vec	256
Examples generated with LSTM-Word2vec and beam search	258
Perplexity over time	259
Using the TensorFlow RNN API	260
Summary	264
Chapter 9: Applications of LSTM – Image Caption Generation	265
Getting to know the data	266
ILSVRC ImageNet dataset	267
The MS-COCO dataset	268
The machine learning pipeline for image caption generation	269
Extracting image features with CNNs	273
Implementation – loading weights and inferencing with VGG-16	274
Building and updating variables	274
Preprocessing inputs	275
Inferring VGG-16	277

Extracting vectorized representations of images	278
Predicting class probabilities with VGG-16	278
Learning word embeddings	280
Preparing captions for feeding into LSTMs	281
Generating data for LSTMs	282
Defining the LSTM	284
Evaluating the results quantitatively	287
BLEU	287
ROUGE	288
METEOR	289
CIDEr	291
BLEU-4 over time for our model	292
Captions generated for test images	293
Using TensorFlow RNN API with pretrained GloVe word vectors	297
Loading GloVe word vectors	298
Cleaning data	299
Using pretrained embeddings with TensorFlow RNN API	302
Defining the pretrained embedding layer and the adaptation layer	303
Defining the LSTM cell and softmax layer	303
Defining inputs and outputs	304
Processing images and text differently	305
Defining the LSTM output calculation	306
Defining the logits and predictions	307
Defining the sequence loss	307
Defining the optimizer	307
Summary	308
Chapter 10: Sequence-to-Sequence Learning – Neural	
Machine Translation	311
Machine translation	312
A brief historical tour of machine translation	313
Rule-based translation	313
Statistical Machine Translation (SMT)	315
Neural Machine Translation (NMT)	317
Understanding Neural Machine Translation	320
Intuition behind NMT	320
NMT architecture	321
The embedding layer	322
The encoder	322
The context vector	323
The decoder	324
Preparing data for the NMT system	325
At training time	325
Reversing the source sentence	326

At testing time	327
Training the NMT	328
Inference with NMT	329
The BLEU score – evaluating the machine translation systems	330
Modified precision	331
Brevity penalty	331
The final BLEU score	332
Implementing an NMT from scratch – a German to English translator	332
Introduction to data	333
Preprocessing data	333
Learning word embeddings	335
Defining the encoder and the decoder	335
Defining the end-to-end output calculation	338
Some translation results	340
Training an NMT jointly with word embeddings	342
Maximizing matchings between the dataset vocabulary and the pretrained embeddings	343
Defining the embeddings layer as a TensorFlow variable	345
Improving NMTs	348
Teacher forcing	348
Deep LSTMs	350
Attention	351
Breaking the context vector bottleneck	351
The attention mechanism in detail	352
Implementing the attention mechanism	356
Defining weights	356
Computing attention	357
Some translation results – NMT with attention	359
Visualizing attention for source and target sentences	361
Other applications of Seq2Seq models – chatbots	363
Training a chatbot	364
Evaluating chatbots – Turing test	365
Summary	366
Chapter 11: Current Trends and the Future of Natural Language Processing	369
Current trends in NLP	370
Word embeddings	370
Region embedding	370
Probabilistic word embedding	374
Ensemble embedding	375
Topic embedding	375
Neural Machine Translation (NMT)	376

Improving the attention mechanism	376
Hybrid MT models	376
Penetration into other research fields	378
Combining NLP with computer vision	378
Visual Question Answering (VQA)	379
Caption generation for images with attention	381
Reinforcement learning	381
Teaching agents to communicate using their own language	382
Dialogue agents with reinforcement learning	383
Generative Adversarial Networks for NLP	384
Towards Artificial General Intelligence	386
One Model to Learn Them All	386
A joint many-task model – growing a neural network for multiple NLP tasks	389
First level – word-based tasks	389
Second level – syntactic tasks	389
Third level – semantic-level tasks	390
NLP for social media	391
Detecting rumors in social media	391
Detecting emotions in social media	391
Analyzing political framing in tweets	393
New tasks emerging	393
Detecting sarcasm	393
Language grounding	394
Skimming text with LSTMs	395
Newer machine learning models	395
Phased LSTM	396
Dilated Recurrent Neural Networks (DRNNs)	397
Summary	398
References	398
Appendix: Mathematical Foundations and Advanced TensorFlow	403
Basic data structures	403
Scalar	403
Vectors	404
Matrices	404
Indexing of a matrix	405
Special types of matrices	406
Identity matrix	406
Diagonal matrix	407
Tensors	407
Tensor/matrix operations	407

Transpose	407
Multiplication	408
Element-wise multiplication	409
Inverse	409
Finding the matrix inverse – Singular Value Decomposition (SVD)	411
Norms	412
Determinant	412
Probability	413
Random variables	413
Discrete random variables	413
Continuous random variables	414
The probability mass/density function	414
Conditional probability	417
Joint probability	417
Marginal probability	417
Bayes' rule	418
Introduction to Keras	418
Introduction to the TensorFlow seq2seq library	421
Defining embeddings for the encoder and decoder	421
Defining the encoder	422
Defining the decoder	422
Visualizing word embeddings with TensorBoard	424
Starting TensorBoard	424
Saving word embeddings and visualizing via TensorBoard	425
Summary	429
Other Books You May Enjoy	431
Index	437
