contents

foreword *xiii*
preface *xv*
acknowledgments *xxi*
about this book *xxiv*
about the authors *xxvii*
about the cover illustration *xxix*

PART 1 WORDY MACHINES 1

1 ***Packets of thought (NLP overview)*** ***3***

1.1 Natural language vs. programming language 4

1.2 The magic 4

Machines that converse *5* ▪ *The math* *6*

1.3 Practical applications 8

1.4 Language through a computer's "eyes" 9

The language of locks *10* ▪ *Regular expressions* *11*
A simple chatbot *12* ▪ *Another way* *16*

1.5 A brief overflight of hyperspace 19

1.6 Word order and grammar 21

1.7 A chatbot natural language pipeline 22

1.8 Processing in depth 25

1.9 Natural language IQ 27

2 ***Build your vocabulary (word tokenization) 30***

2.1 Challenges (a preview of stemming) 32

2.2 Building your vocabulary with a tokenizer 33

Dot product 41 ▪ Measuring bag-of-words overlap 42
A token improvement 43 ▪ Extending your vocabulary with n-grams 48 ▪ Normalizing your vocabulary 54

2.3 Sentiment 62

VADER—A rule-based sentiment analyzer 64 ▪ Naive Bayes 65

3 ***Math with words (TF-IDF vectors) 70***

3.1 Bag of words 71

3.2 Vectorizing 76

Vector spaces 79

3.3 Zipf's Law 83

3.4 Topic modeling 86

Return of Zipf 89 ▪ Relevance ranking 90 ▪ Tools 93
Alternatives 93 ▪ Okapi BM25 95 ▪ What's next 95

4 ***Finding meaning in word counts (semantic analysis) 97***

4.1 From word counts to topic scores 98

TF-IDF vectors and lemmatization 99 ▪ Topic vectors 99
Thought experiment 101 ▪ An algorithm for scoring topics 105
An LDA classifier 107

4.2 Latent semantic analysis 111

Your thought experiment made real 113

4.3 Singular value decomposition 116

U—left singular vectors 118 ▪ S—singular values 119
VT—right singular vectors 120 ▪ SVD matrix orientation 120
Truncating the topics 121

4.4 Principal component analysis 123

PCA on 3D vectors 125 ▪ Stop horsing around and get back to NLP 126 ▪ Using PCA for SMS message semantic analysis 128
Using truncated SVD for SMS message semantic analysis 130
How well does LSA work for spam classification? 131

4.5 Latent Dirichlet allocation (LDiA) 134

The LDiA idea 135 ▪ LDiA topic model for SMS messages 137
LDiA + LDA = spam classifier 140 ▪ A fairer comparison: 32 LDiA topics 142

4.6 Distance and similarity 143

4.7 Steering with feedback 146

Linear discriminant analysis 147

4.8 Topic vector power 148

Semantic search 150 ▪ Improvements 152

PART 2 DEEPER LEARNING (NEURAL NETWORKS) 153

5 ***Baby steps with neural networks (perceptrons and backpropagation) 155***

5.1 Neural networks, the ingredient list 156

Perceptron 157 ▪ A numerical perceptron 157 ▪ Detour through bias 158 ▪ Let's go skiing—the error surface 172 Off the chair lift, onto the slope 173 ▪ Let's shake things up a bit 174 ▪ Keras: neural networks in Python 175 ▪ Onward and deepward 179 ▪ Normalization: input with style 179

6 ***Reasoning with word vectors (Word2vec) 181***

6.1 Semantic queries and analogies 182

Analogy questions 183

6.2 Word vectors 184

Vector-oriented reasoning 187 ▪ How to compute Word2vec representations 191 ▪ How to use the gensim.word2vec module 200 ▪ How to generate your own word vector representations 202 ▪ Word2vec vs. GloVe (Global Vectors) 205 fastText 205 ▪ Word2vec vs. LSA 206 ▪ Visualizing word relationships 207 ▪ Unnatural words 214 ▪ Document similarity with Doc2vec 215

7 ***Getting words in order with convolutional neural networks (CNNs) 218***

7.1 Learning meaning 220

7.2 Toolkit 221

7.3 Convolutional neural nets 222

Building blocks 223 ▪ Step size (stride) 224 ▪ Filter composition 224 ▪ Padding 226 ▪ Learning 228

7.4 Narrow windows indeed 228

Implementation in Keras: prepping the data 230 ▪ Convolutional neural network architecture 235 ▪ Pooling 236 Dropout 238 ▪ The cherry on the sundae 239 ▪ Let's get to

learning (training) 241 ▪ Using the model in a pipeline 243
Where do you go from here? 244

8 ***Loopy (recurrent) neural networks (RNNs) 247***

8.1 Remembering with recurrent networks 250

Backpropagation through time 255 ▪ When do we update what? 257 ▪ Recap 259 ▪ There's always a catch 259
Recurrent neural net with Keras 260

8.2 Putting things together 264

8.3 Let's get to learning our past selves 266

8.4 Hyperparameters 267

8.5 Predicting 269

Statefulness 270 ▪ Two-way street 271 ▪ What is this thing? 272

9 ***Improving retention with long short-term memory networks 274***

9.1 LSTM 275

Backpropagation through time 284 ▪ Where does the rubber hit the road? 287 ▪ Dirty data 288 ▪ Back to the dirty data 291
Words are hard. Letters are easier. 292 ▪ My turn to chat 298
My turn to speak more clearly 300 ▪ Learned how to say, but not yet what 308 ▪ Other kinds of memory 308 ▪ Going deeper 309

10 ***Sequence-to-sequence models and attention 311***

10.1 Encoder-decoder architecture 312

Decoding thought 313 ▪ Look familiar? 315 ▪ Sequence-to-sequence conversation 316 ▪ LSTM review 317

10.2 Assembling a sequence-to-sequence pipeline 318

Preparing your dataset for the sequence-to-sequence training 318
Sequence-to-sequence model in Keras 320 ▪ Sequence encoder 320 ▪ Thought decoder 322 ▪ Assembling the sequence-to-sequence network 323

10.3 Training the sequence-to-sequence network 324

Generate output sequences 325

10.4 Building a chatbot using sequence-to-sequence networks 326

Preparing the corpus for your training 326 ▪ Building your character dictionary 327 ▪ Generate one-hot encoded training sets 328 ▪ Train your sequence-to-sequence chatbot 329

Assemble the model for sequence generation 330 ▪ Predicting a sequence 330 ▪ Generating a response 331 ▪ Converse with your chatbot 331

10.5 Enhancements 332

Reduce training complexity with bucketing 332 ▪ Paying attention 333

10.6 In the real world 334

PART 3 GETTING REAL (REAL-WORLD NLP CHALLENGES) 337

11 *Information extraction (named entity extraction and question answering) 339*

11.1 Named entities and relations 339

A knowledge base 340 ▪ Information extraction 343

11.2 Regular patterns 343

Regular expressions 344 ▪ Information extraction as ML feature extraction 345

11.3 Information worth extracting 346

Extracting GPS locations 347 ▪ Extracting dates 347

11.4 Extracting relationships (relations) 352

Part-of-speech (POS) tagging 353 ▪ Entity name normalization 357 Relation normalization and extraction 358 ▪ Word patterns 358 Segmentation 359 ▪ Why won't split('.!?') work? 360 Sentence segmentation with regular expressions 361

11.5 In the real world 363

12 *Getting chatty (dialog engines) 365*

12.1 Language skill 366

Modern approaches 367 ▪ A hybrid approach 373

12.2 Pattern-matching approach 373

A pattern-matching chatbot with AIML 375 ▪ A network view of pattern matching 381

12.3 Grounding 382

12.4 Retrieval (search) 384

The context challenge 384 ▪ Example retrieval-based chatbot 386 ▪ A search-based chatbot 389

12.5 Generative models 391

Chat about NLPIA 392 ▪ Pros and cons of each approach 394

12.6 Four-wheel drive 395

The Will to succeed 395

12.7 Design process 396

12.8 Trickery 399

Ask questions with predictable answers 399 ▪ Be entertaining 399 When all else fails, search 400 ▪ Being popular 400 ▪ Be a connector 400 ▪ Getting emotional 400

12.9 In the real world 401

13 ***Scaling up (optimization, parallelization, and batch processing) 403***

13.1 Too much of a good thing (data) 404

13.2 Optimizing NLP algorithms 404

Indexing 405 ▪ Advanced indexing 406 ▪ Advanced indexing with Annoy 408 ▪ Why use approximate indexes at all? 412 An indexing workaround: discretizing 413

13.3 Constant RAM algorithms 414

Gensim 414 ▪ Graph computing 415

13.4 Parallelizing your NLP computations 416

Training NLP models on GPUs 416 ▪ Renting vs. buying 417 GPU rental options 418 ▪ Tensor processing units 419

13.5 Reducing the memory footprint during model training 419

13.6 Gaining model insights with TensorBoard 422

How to visualize word embeddings 423

appendix A Your NLP tools 427

appendix B Playful Python and regular expressions 434

appendix C Vectors and matrices (linear algebra fundamentals) 440

appendix D Machine learning tools and techniques 446

appendix E Setting up your AWS GPU 459

appendix F Locality sensitive hashing 473

resources 481

glossary 490

index 497