Natural Language Processing

Jacob Eisenstein

November 13, 2018

Contents

Contents 1

Preface i

Background i

How to use this book ii

1 Introduction 1

1.1 Natural language processing and its neighbors 1

1.2 Three themes in natural language processing 6

1.2.1 Learning and knowledge 6

1.2.2 Search and learning 7

1.2.3 Relational, compositional, and distributional perspectives 9

I Learning 11

2 Linear text classification 13

2.1 The bag of words 13

2.2 Naïve Bayes 17

2.2.1 Types and tokens 19

2.2.2 Prediction 20

2.2.3 Estimation 21

2.2.4 Smoothing 22

2.2.5 Setting hyperparameters 23

2.3 Discriminative learning 24

2.3.1 Perceptron 25

2.3.2 Averaged perceptron 27

2.4 Loss functions and large-margin classification 27

2.4.1 Online large margin classification 30

2.4.2 *Derivation of the online support vector machine 32

2.5 Logistic regression 35

2.5.1 Regularization 36
2.5.2 Gradients 37
2.6 Optimization 37
2.6.1 Batch optimization 38
2.6.2 Online optimization 39
2.7 *Additional topics in classification 41
2.7.1 Feature selection by regularization 41
2.7.2 Other views of logistic regression 41
2.8 Summary of learning algorithms 43

3 Nonlinear classification 47
3.1 Feedforward neural networks 48
3.2 Designing neural networks 50
3.2.1 Activation functions 50
3.2.2 Network structure 51
3.2.3 Outputs and loss functions 52
3.2.4 Inputs and lookup layers 53
3.3 Learning neural networks 53
3.3.1 Backpropagation 55
3.3.2 Regularization and dropout 57
3.3.3 *Learning theory 58
3.3.4 Tricks 59
3.4 Convolutional neural networks 62

4 Linguistic applications of classification 69
4.1 Sentiment and opinion analysis 69
4.1.1 Related problems 70
4.1.2 Alternative approaches to sentiment analysis 72
4.2 Word sense disambiguation 73
4.2.1 How many word senses? 74
4.2.2 Word sense disambiguation as classification 75
4.3 Design decisions for text classification 76
4.3.1 What is a word? 76
4.3.2 How many words? 79
4.3.3 Count or binary? 80
4.4 Evaluating classifiers 80
4.4.1 Precision, recall, and F-MEASURE 81
4.4.2 Threshold-free metrics 83
4.4.3 Classifier comparison and statistical significance 84
4.4.4 *Multiple comparisons 87
4.5 Building datasets 88

4.5.1 Metadata as labels 88
4.5.2 Labeling data 88

5 Learning without supervision 95
5.1 Unsupervised learning 95
5.1.1 K-means clustering 96
5.1.2 Expectation-Maximization (EM) 98
5.1.3 EM as an optimization algorithm 102
5.1.4 How many clusters? 103
5.2 Applications of expectation-maximization 104
5.2.1 Word sense induction 104
5.2.2 Semi-supervised learning 105
5.2.3 Multi-component modeling 106
5.3 Semi-supervised learning 107
5.3.1 Multi-view learning 108
5.3.2 Graph-based algorithms 109
5.4 Domain adaptation 110
5.4.1 Supervised domain adaptation 111
5.4.2 Unsupervised domain adaptation 112
5.5 *Other approaches to learning with latent variables 114
5.5.1 Sampling 115
5.5.2 Spectral learning 117

II Sequences and trees 123

6 Language models 125
6.1 N-gram language models 126
6.2 Smoothing and discounting 129
6.2.1 Smoothing 129
6.2.2 Discounting and backoff 130
6.2.3 *Interpolation 131
6.2.4 *Kneser-Ney smoothing 133
6.3 Recurrent neural network language models 133
6.3.1 Backpropagation through time 136
6.3.2 Hyperparameters 137
6.3.3 Gated recurrent neural networks 137
6.4 Evaluating language models 139
6.4.1 Held-out likelihood 139
6.4.2 Perplexity 140
6.5 Out-of-vocabulary words 141

Under contract with MIT Press, shared under CC-BY-NC-ND license.

7 Sequence labeling 145
7.1 Sequence labeling as classification . . . 145
7.2 Sequence labeling as structure prediction . . . 147
7.3 The Viterbi algorithm . . . 149
7.3.1 Example . . . 152
7.3.2 Higher-order features . . . 153
7.4 Hidden Markov Models . . . 153
7.4.1 Estimation . . . 155
7.4.2 Inference . . . 155
7.5 Discriminative sequence labeling with features . . . 157
7.5.1 Structured perceptron . . . 160
7.5.2 Structured support vector machines . . . 160
7.5.3 Conditional random fields . . . 162
7.6 Neural sequence labeling . . . 167
7.6.1 Recurrent neural networks . . . 167
7.6.2 Character-level models . . . 169
7.6.3 Convolutional Neural Networks for Sequence Labeling . . . 170
7.7 *Unsupervised sequence labeling . . . 170
7.7.1 Linear dynamical systems . . . 172
7.7.2 Alternative unsupervised learning methods . . . 172
7.7.3 Semiring notation and the generalized viterbi algorithm . . . 172

8 Applications of sequence labeling 175
8.1 Part-of-speech tagging . . . 175
8.1.1 Parts-of-Speech . . . 176
8.1.2 Accurate part-of-speech tagging . . . 180
8.2 Morphosyntactic Attributes . . . 182
8.3 Named Entity Recognition . . . 183
8.4 Tokenization . . . 185
8.5 Code switching . . . 186
8.6 Dialogue acts . . . 187

9 Formal language theory 191
9.1 Regular languages . . . 192
9.1.1 Finite state acceptors . . . 193
9.1.2 Morphology as a regular language . . . 194
9.1.3 Weighted finite state acceptors . . . 196
9.1.4 Finite state transducers . . . 201
9.1.5 *Learning weighted finite state automata . . . 206
9.2 Context-free languages . . . 207
9.2.1 Context-free grammars . . . 208

9.2.2 Natural language syntax as a context-free language 211
9.2.3 A phrase-structure grammar for English 213
9.2.4 Grammatical ambiguity 218
9.3 *Mildly context-sensitive languages 218
9.3.1 Context-sensitive phenomena in natural language 219
9.3.2 Combinatory categorial grammar 220

10 Context-free parsing 225
10.1 Deterministic bottom-up parsing 226
10.1.1 Recovering the parse tree 227
10.1.2 Non-binary productions 227
10.1.3 Complexity 229
10.2 Ambiguity 229
10.2.1 Parser evaluation 230
10.2.2 Local solutions 231
10.3 Weighted Context-Free Grammars 232
10.3.1 Parsing with weighted context-free grammars 234
10.3.2 Probabilistic context-free grammars 235
10.3.3 *Semiring weighted context-free grammars 237
10.4 Learning weighted context-free grammars 238
10.4.1 Probabilistic context-free grammars 238
10.4.2 Feature-based parsing 239
10.4.3 *Conditional random field parsing 240
10.4.4 Neural context-free grammars 242
10.5 Grammar refinement 242
10.5.1 Parent annotations and other tree transformations 243
10.5.2 Lexicalized context-free grammars 244
10.5.3 *Refinement grammars 248
10.6 Beyond context-free parsing 249
10.6.1 Reranking 250
10.6.2 Transition-based parsing 251

11 Dependency parsing 257
11.1 Dependency grammar 257
11.1.1 Heads and dependents 258
11.1.2 Labeled dependencies 259
11.1.3 Dependency subtrees and constituents 260
11.2 Graph-based dependency parsing 262
11.2.1 Graph-based parsing algorithms 264
11.2.2 Computing scores for dependency arcs 265
11.2.3 Learning 267

Under contract with MIT Press, shared under CC-BY-NC-ND license.

11.3 Transition-based dependency parsing . . . 268
11.3.1 Transition systems for dependency parsing . . . 269
11.3.2 Scoring functions for transition-based parsers . . . 273
11.3.3 Learning to parse . . . 274
11.4 Applications . . . 277

III Meaning 283

12 Logical semantics 285
12.1 Meaning and denotation . . . 286
12.2 Logical representations of meaning . . . 287
12.2.1 Propositional logic . . . 287
12.2.2 First-order logic . . . 288
12.3 Semantic parsing and the lambda calculus . . . 291
12.3.1 The lambda calculus . . . 292
12.3.2 Quantification . . . 293
12.4 Learning semantic parsers . . . 296
12.4.1 Learning from derivations . . . 297
12.4.2 Learning from logical forms . . . 299
12.4.3 Learning from denotations . . . 301

13 Predicate-argument semantics 305
13.1 Semantic roles . . . 307
13.1.1 VerbNet . . . 308
13.1.2 Proto-roles and PropBank . . . 309
13.1.3 FrameNet . . . 310
13.2 Semantic role labeling . . . 312
13.2.1 Semantic role labeling as classification . . . 312
13.2.2 Semantic role labeling as constrained optimization . . . 315
13.2.3 Neural semantic role labeling . . . 317
13.3 Abstract Meaning Representation . . . 318
13.3.1 AMR Parsing . . . 321

14 Distributional and distributed semantics 325
14.1 The distributional hypothesis . . . 325
14.2 Design decisions for word representations . . . 327
14.2.1 Representation . . . 327
14.2.2 Context . . . 328
14.2.3 Estimation . . . 329
14.3 Latent semantic analysis . . . 329

14.4 Brown clusters 331
14.5 Neural word embeddings 334
14.5.1 Continuous bag-of-words (CBOW) 334
14.5.2 Skipgrams 335
14.5.3 Computational complexity 335
14.5.4 Word embeddings as matrix factorization 337
14.6 Evaluating word embeddings 338
14.6.1 Intrinsic evaluations 339
14.6.2 Extrinsic evaluations 339
14.6.3 Fairness and bias 340
14.7 Distributed representations beyond distributional statistics 341
14.7.1 Word-internal structure 341
14.7.2 Lexical semantic resources 343
14.8 Distributed representations of multiword units 344
14.8.1 Purely distributional methods 344
14.8.2 Distributional-compositional hybrids 345
14.8.3 Supervised compositional methods 346
14.8.4 Hybrid distributed-symbolic representations 346

15 Reference Resolution 351
15.1 Forms of referring expressions 352
15.1.1 Pronouns 352
15.1.2 Proper Nouns 357
15.1.3 Nominals 357
15.2 Algorithms for coreference resolution 358
15.2.1 Mention-pair models 359
15.2.2 Mention-ranking models 360
15.2.3 Transitive closure in mention-based models 361
15.2.4 Entity-based models 362
15.3 Representations for coreference resolution 367
15.3.1 Features 367
15.3.2 Distributed representations of mentions and entities 370
15.4 Evaluating coreference resolution 373

16 Discourse 379
16.1 Segments 379
16.1.1 Topic segmentation 380
16.1.2 Functional segmentation 381
16.2 Entities and reference 381
16.2.1 Centering theory 382
16.2.2 The entity grid 383

Under contract with MIT Press, shared under CC-BY-NC-ND license.

16.2.3 *Formal semantics beyond the sentence level 384
16.3 Relations 385
16.3.1 Shallow discourse relations 385
16.3.2 Hierarchical discourse relations 389
16.3.3 Argumentation 392
16.3.4 Applications of discourse relations 393

IV Applications 401

17 Information extraction 403
17.1 Entities 405
17.1.1 Entity linking by learning to rank 406
17.1.2 Collective entity linking 408
17.1.3 *Pairwise ranking loss functions 409
17.2 Relations 411
17.2.1 Pattern-based relation extraction 412
17.2.2 Relation extraction as a classification task 413
17.2.3 Knowledge base population 416
17.2.4 Open information extraction 419
17.3 Events 420
17.4 Hedges, denials, and hypotheticals 422
17.5 Question answering and machine reading 424
17.5.1 Formal semantics 424
17.5.2 Machine reading 425

18 Machine translation 431
18.1 Machine translation as a task 431
18.1.1 Evaluating translations 433
18.1.2 Data 435
18.2 Statistical machine translation 436
18.2.1 Statistical translation modeling 437
18.2.2 Estimation 438
18.2.3 Phrase-based translation 439
18.2.4 *Syntax-based translation 441
18.3 Neural machine translation 442
18.3.1 Neural attention 444
18.3.2 *Neural machine translation without recurrence 446
18.3.3 Out-of-vocabulary words 448
18.4 Decoding 449
18.5 Training towards the evaluation metric 451

19 Text generation **457**
19.1 Data-to-text generation 457
19.1.1 Latent data-to-text alignment 459
19.1.2 Neural data-to-text generation 460
19.2 Text-to-text generation 464
19.2.1 Neural abstractive summarization 464
19.2.2 Sentence fusion for multi-document summarization 465
19.3 Dialogue 466
19.3.1 Finite-state and agenda-based dialogue systems 467
19.3.2 Markov decision processes 468
19.3.3 Neural chatbots 470

A Probability **475**
A.1 Probabilities of event combinations 475
A.1.1 Probabilities of disjoint events 476
A.1.2 Law of total probability 477
A.2 Conditional probability and Bayes' rule 477
A.3 Independence 479
A.4 Random variables 480
A.5 Expectations 481
A.6 Modeling and estimation 482

B Numerical optimization **485**
B.1 Gradient descent 486
B.2 Constrained optimization 486
B.3 Example: Passive-aggressive online learning 487

Bibliography **489**

Under contract with MIT Press, shared under CC-BY-NC-ND license.

Preface

The goal of this text is focus on a core subset of the natural language processing, unified by the concepts of learning and search. A remarkable number of problems in natural language processing can be solved by a compact set of methods:

Search. Viterbi, CKY, minimum spanning tree, shift-reduce, integer linear programming, beam search.

Learning. Maximum-likelihood estimation, logistic regression, perceptron, expectation-maximization, matrix factorization, backpropagation.

This text explains how these methods work, and how they can be applied to a wide range of tasks: document classification, word sense disambiguation, part-of-speech tagging, named entity recognition, parsing, coreference resolution, relation extraction, discourse analysis, language modeling, and machine translation.

Background

Because natural language processing draws on many different intellectual traditions, almost everyone who approaches it feels underprepared in one way or another. Here is a summary of what is expected, and where you can learn more:

Mathematics and machine learning. The text assumes a background in multivariate calculus and linear algebra: vectors, matrices, derivatives, and partial derivatives. You should also be familiar with probability and statistics. A review of basic probability is found in Appendix A, and a minimal review of numerical optimization is found in Appendix B. For linear algebra, the online course and textbook from Strang (2016) provide an excellent review. Deisenroth et al. (2018) are currently preparing a textbook on *Mathematics for Machine Learning*, a draft can be found online.[1] For an introduction to probabilistic modeling and estimation, see James et al. (2013); for

[1] `https://mml-book.github.io/`

a more advanced and comprehensive discussion of the same material, the classic reference is Hastie et al. (2009).

Linguistics. This book assumes no formal training in linguistics, aside from elementary concepts likes nouns and verbs, which you have probably encountered in the study of English grammar. Ideas from linguistics are introduced throughout the text as needed, including discussions of morphology and syntax (chapter 9), semantics (chapters 12 and 13), and discourse (chapter 16). Linguistic issues also arise in the application-focused chapters 4, 8, and 18. A short guide to linguistics for students of natural language processing is offered by Bender (2013); you are encouraged to start there, and then pick up a more comprehensive introductory textbook (e.g., Akmajian et al., 2010; Fromkin et al., 2013).

Computer science. The book is targeted at computer scientists, who are assumed to have taken introductory courses on the analysis of algorithms and complexity theory. In particular, you should be familiar with asymptotic analysis of the time and memory costs of algorithms, and with the basics of dynamic programming. The classic text on algorithms is offered by Cormen et al. (2009); for an introduction to the theory of computation, see Arora and Barak (2009) and Sipser (2012).

How to use this book

After the introduction, the textbook is organized into four main units:

Learning. This section builds up a set of machine learning tools that will be used throughout the other sections. Because the focus is on machine learning, the text representations and linguistic phenomena are mostly simple: "bag-of-words" text classification is treated as a model example. Chapter 4 describes some of the more linguistically interesting applications of word-based text analysis.

Sequences and trees. This section introduces the treatment of language as a structured phenomena. It describes sequence and tree representations and the algorithms that they facilitate, as well as the limitations that these representations impose. Chapter 9 introduces finite state automata and briefly overviews a context-free account of English syntax.

Meaning. This section takes a broad view of efforts to represent and compute meaning from text, ranging from formal logic to neural word embeddings. It also includes two topics that are closely related to semantics: resolution of ambiguous references, and analysis of multi-sentence discourse structure.

Applications. The final section offers chapter-length treatments on three of the most prominent applications of natural language processing: information extraction, machine translation, and text generation. Each of these applications merits a textbook length treatment of its own (Koehn, 2009; Grishman, 2012; Reiter and Dale, 2000); the chapters here explain some of the most well known systems using the formalisms and methods built up earlier in the book, while introducing methods such as neural attention.

Each chapter contains some advanced material, which is marked with an asterisk. This material can be safely omitted without causing misunderstandings later on. But even without these advanced sections, the text is too long for a single semester course, so instructors will have to pick and choose among the chapters.

Chapters 1-3 provide building blocks that will be used throughout the book, and chapter 4 describes some critical aspects of the practice of language technology. Language models (chapter 6), sequence labeling (chapter 7), and parsing (chapter 10 and 11) are canonical topics in natural language processing, and distributed word embeddings (chapter 14) have become ubiquitous. Of the applications, machine translation (chapter 18) is the best choice: it is more cohesive than information extraction, and more mature than text generation. Many students will benefit from the review of probability in Appendix A.

- A course focusing on machine learning should add the chapter on unsupervised learning (chapter 5). The chapters on predicate-argument semantics (chapter 13), reference resolution (chapter 15), and text generation (chapter 19) are particularly influenced by recent progress in machine learning, including deep neural networks and learning to search.
- A course with a more linguistic orientation should add the chapters on applications of sequence labeling (chapter 8), formal language theory (chapter 9), semantics (chapter 12 and 13), and discourse (chapter 16).
- For a course with a more applied focus, I recommend the chapters on applications of sequence labeling (chapter 8), predicate-argument semantics (chapter 13), information extraction (chapter 17), and text generation (chapter 19).

Acknowledgments

Several colleagues, students, and friends read early drafts of chapters in their areas of expertise, including Yoav Artzi, Kevin Duh, Heng Ji, Jessy Li, Brendan O'Connor, Yuval Pinter, Shawn Ling Ramirez, Nathan Schneider, Pamela Shapiro, Noah A. Smith, Sandeep Soni, and Luke Zettlemoyer. I also thank the anonymous reviewers, particularly reviewer 4, who provided detailed line-by-line edits and suggestions. The text benefited from high-level discussions with my editor Marie Lufkin Lee, as well as Kevin Murphy, Shawn Ling Ramirez, and Bonnie Webber. In addition, there are many students, colleagues, friends, and family who found mistakes in early drafts, or who recommended key references.

Under contract with MIT Press, shared under CC-BY-NC-ND license.

These include: Parminder Bhatia, Kimberly Caras, Jiahao Cai, Justin Chen, Rodolfo Delmonte, Murtaza Dhuliawala, Yantao Du, Barbara Eisenstein, Luiz C. F. Ribeiro, Chris Gu, Joshua Killingsworth, Jonathan May, Taha Merghani, Gus Monod, Raghavendra Murali, Nidish Nair, Brendan O'Connor, Dan Oneata, Brandon Peck, Yuval Pinter, Nathan Schneider, Jianhao Shen, Zhewei Sun, Rubin Tsui, Ashwin Cunnapakkam Vinjimur, Denny Vrandečić, William Yang Wang, Clay Washington, Ishan Waykul, Aobo Yang, Xavier Yao, Yuyu Zhang, and several anonymous commenters. Clay Washington tested some of the programming exercises, and Varun Gupta tested some of the written exercises. Thanks to Kelvin Xu for sharing a high-resolution version of Figure 19.3.

Most of the book was written while I was at Georgia Tech's School of Interactive Computing. I thank the School for its support of this project, and I thank my colleagues there for their help and support at the beginning of my faculty career. I also thank (and apologize to) the many students in Georgia Tech's CS 4650 and 7650 who suffered through early versions of the text. The book is dedicated to my parents.

Notation

As a general rule, words, word counts, and other types of observations are indicated with Roman letters (a, b, c); parameters are indicated with Greek letters (α, β, θ). Vectors are indicated with bold script for both random variables $\boldsymbol{x}$ and parameters $\boldsymbol{\theta}$. Other useful notations are indicated in the table below.

Basics	
$\exp x$	the base-2 exponent, 2^x
$\log x$	the base-2 logarithm, $\log_2 x$
$\{x_n\}_{n=1}^N$	the set $\{x_1, x_2, \ldots, x_N\}$
x_i^j	x_i raised to the power j
$x_i^{(j)}$	indexing by both i and j

Linear algebra	
$\boldsymbol{x}^{(i)}$	a column vector of feature counts for instance i, often word counts
$\boldsymbol{x}_{j:k}$	elements j through k (inclusive) of a vector $\boldsymbol{x}$
$[\boldsymbol{x}; \boldsymbol{y}]$	vertical concatenation of two column vectors
$[\boldsymbol{x}, \boldsymbol{y}]$	horizontal concatenation of two column vectors
$\boldsymbol{e}_n$	a "one-hot" vector with a value of 1 at position n, and zero everywhere else
$\boldsymbol{\theta}^\top$	the transpose of a column vector $\boldsymbol{\theta}$
$\boldsymbol{\theta} \cdot \boldsymbol{x}^{(i)}$	the dot product $\sum_{j=1}^N \theta_j \times x_j^{(i)}$
$\mathbf{X}$	a matrix
$x_{i,j}$	row i, column j of matrix $\mathbf{X}$
$\text{Diag}(\boldsymbol{x})$	a matrix with $\boldsymbol{x}$ on the diagonal, e.g., $\begin{pmatrix} x_1 & 0 & 0 \\ 0 & x_2 & 0 \\ 0 & 0 & x_3 \end{pmatrix}$
$\mathbf{X}^{-1}$	the inverse of matrix $\mathbf{X}$

Text datasets	
w_m	word token at position m
N	number of training instances
M	length of a sequence (of words or tags)
V	number of words in vocabulary
$y^{(i)}$	the true label for instance i
$\hat{y}$	a predicted label
$\mathcal{Y}$	the set of all possible labels
K	number of possible labels $K = \lvert\mathcal{Y}\rvert$
$\square$	the start token
$\blacksquare$	the stop token
$\boldsymbol{y}^{(i)}$	a structured label for instance i, such as a tag sequence
$\mathcal{Y}(\boldsymbol{w})$	the set of possible labelings for the word sequence $\boldsymbol{w}$
$\Diamond$	the start tag
$\blacklozenge$	the stop tag

Probabilities	
$\Pr(A)$	probability of event A
$\Pr(A \mid B)$	probability of event A, conditioned on event B
$\mathrm{p}_B(b)$	the marginal probability of random variable B taking value b; written $\mathrm{p}(b)$ when the choice of random variable is clear from context
$\mathrm{p}_{B\mid A}(b \mid a)$	the probability of random variable B taking value b, conditioned on A taking value a; written $\mathrm{p}(b \mid a)$ when clear from context
$A \sim p$	the random variable A is distributed according to distribution p. For example, $X \sim \mathcal{N}(0, 1)$ states that the random variable X is drawn from a normal distribution with zero mean and unit variance.
$A \mid B \sim p$	conditioned on the random variable B, A is distributed according to p.[2]

Machine learning	
$\Psi(\boldsymbol{x}^{(i)}, y)$	the score for assigning label y to instance i
$\boldsymbol{f}(\boldsymbol{x}^{(i)}, y)$	the feature vector for instance i with label y
$\boldsymbol{\theta}$	a (column) vector of weights
$\ell^{(i)}$	loss on an individual instance i
L	objective function for an entire dataset
$\mathcal{L}$	log-likelihood of a dataset
λ	the amount of regularization