Natural Language Processing

Jacob Eisenstein

November 13, 2018

Contents

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.

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

Chapter 1

Introduction

Natural language processing is the set of methods for making human language accessible to computers. In the past decade, natural language processing has become embedded in our daily lives: automatic machine translation is ubiquitous on the web and in social media; text classification keeps our email inboxes from collapsing under a deluge of spam; search engines have moved beyond string matching and network analysis to a high degree of linguistic sophistication; dialog systems provide an increasingly common and effective way to get and share information.

These diverse applications are based on a common set of ideas, drawing on algorithms, linguistics, logic, statistics, and more. The goal of this text is to provide a survey of these foundations. The technical fun starts in the next chapter; the rest of this current chapter situates natural language processing with respect to other intellectual disciplines, identifies some high-level themes in contemporary natural language processing, and advises the reader on how best to approach the subject.

1.1 Natural language processing and its neighbors

Natural language processing draws on many other intellectual traditions, from formal linguistics to statistical physics. This section briefly situates natural language processing with respect to some of its closest neighbors.

Computational Linguistics Most of the meetings and journals that host natural language processing research bear the name "computational linguistics", and the terms may be thought of as essentially synonymous. But while there is substantial overlap, there is an important difference in focus. In linguistics, language is the object of study. Computational methods may be brought to bear, just as in scientific disciplines like computational biology and computational astronomy, but they play only a supporting role. In contrast, natural language processing is focused on the design and analysis of computational algorithms and representations for processing natural human language. The goal of natural language processing is to provide new computational capabilities around human language: for example, extracting information from texts, translating between languages, answering questions, holding a conversation, taking instructions, and so on. Fundamental linguistic insights may be crucial for accomplishing these tasks, but success is ultimately measured by whether and how well the job gets done.

Machine Learning Contemporary approaches to natural language processing rely heavily on machine learning, which makes it possible to build complex computer programs from examples. Machine learning provides an array of general techniques for tasks like converting a sequence of discrete tokens in one vocabulary to a sequence of discrete tokens in another vocabulary — a generalization of what one might informally call "translation." Much of today's natural language processing research can be thought of as applied machine learning. However, natural language processing has characteristics that distinguish it from many of machine learning's other application domains.

- Unlike images or audio, text data is fundamentally discrete, with meaning created by combinatorial arrangements of symbolic units. This is particularly consequential for applications in which text is the output, such as translation and summarization, because it is not possible to gradually approach an optimal solution.
- Although the set of words is discrete, new words are always being created. Furthermore, the distribution over words (and other linguistic elements) resembles that of a **power law**[1] (Zipf, 1949): there will be a few words that are very frequent, and a long tail of words that are rare. A consequence is that natural language processing algorithms must be especially robust to observations that do not occur in the training data.
- Language is **compositional**: units such as words can combine to create phrases, which can combine by the very same principles to create larger phrases. For example, a **noun phrase** can be created by combining a smaller noun phrase with a **prepositional phrase**, as in *the whiteness of the whale*. The prepositional phrase is created by combining a preposition (in this case, *of*) with another noun phrase (*the whale*). In this way, it is possible to create arbitrarily long phrases, such as,

  (1.1) ...huge globular pieces of the whale of the bigness of a human head.[2]

  The meaning of such a phrase must be analyzed in accord with the underlying hierarchical structure. In this case, *huge globular pieces of the whale* acts as a single noun

[1] Throughout the text, **boldface** will be used to indicate keywords that appear in the index.

[2] Throughout the text, this notation will be used to introduce linguistic examples.

> phrase, which is conjoined with the prepositional phrase *of the bigness of a human head*. The interpretation would be different if instead, *huge globular pieces* were conjoined with the prepositional phrase *of the whale of the bigness of a human head* — implying a disappointingly small whale. Even though text appears as a sequence, machine learning methods must account for its implicit recursive structure.

Artificial Intelligence The goal of artificial intelligence is to build software and robots with the same range of abilities as humans (Russell and Norvig, 2009). Natural language processing is relevant to this goal in several ways. On the most basic level, the capacity for language is one of the central features of human intelligence, and is therefore a prerequisite for artificial intelligence.[3] Second, much of artificial intelligence research is dedicated to the development of systems that can reason from premises to a conclusion, but such algorithms are only as good as what they know (Dreyfus, 1992). Natural language processing is a potential solution to the "knowledge bottleneck", by acquiring knowledge from texts, and perhaps also from conversations. This idea goes all the way back to Turing's 1949 paper *Computing Machinery and Intelligence*, which proposed the **Turing test** for determining whether artificial intelligence had been achieved (Turing, 2009).

Conversely, reasoning is sometimes essential for basic tasks of language processing, such as resolving a pronoun. **Winograd schemas** are examples in which a single word changes the likely referent of a pronoun, in a way that seems to require knowledge and reasoning to decode (Levesque et al., 2011). For example,

(1.2) The trophy doesn't fit into the brown suitcase because **it** is too [small/large].

When the final word is *small*, then the pronoun *it* refers to the suitcase; when the final word is *large*, then *it* refers to the trophy. Solving this example requires spatial reasoning; other schemas require reasoning about actions and their effects, emotions and intentions, and social conventions.

Such examples demonstrate that natural language understanding cannot be achieved in isolation from knowledge and reasoning. Yet the history of artificial intelligence has been one of increasing specialization: with the growing volume of research in subdisciplines such as natural language processing, machine learning, and computer vision, it is

[3] This view is shared by some, but not all, prominent researchers in artificial intelligence. Michael Jordan, a specialist in machine learning, has said that if he had a billion dollars to spend on any large research project, he would spend it on natural language processing (`https://www.reddit.com/r/MachineLearning/comments/2fxi6v/ama_michael_i_jordan/`). On the other hand, in a public discussion about the future of artificial intelligence in February 2018, computer vision researcher Yann Lecun argued that despite its many practical applications, language is perhaps "number 300" in the priority list for artificial intelligence research, and that it would be a great achievement if AI could attain the capabilities of an orangutan, which do not include language (`http://www.abigailsee.com/2018/02/21/deep-learning-structure-and-innate-priors.html`).

difficult for anyone to maintain expertise across the entire field. Still, recent work has demonstrated interesting connections between natural language processing and other areas of AI, including computer vision (e.g., Antol et al., 2015) and game playing (e.g., Branavan et al., 2009). The dominance of machine learning throughout artificial intelligence has led to a broad consensus on representations such as graphical models and computation graphs, and on algorithms such as backpropagation and combinatorial optimization. Many of the algorithms and representations covered in this text are part of this consensus.

Computer Science The discrete and recursive nature of natural language invites the application of theoretical ideas from computer science. Linguists such as Chomsky and Montague have shown how formal language theory can help to explain the syntax and semantics of natural language. Theoretical models such as finite-state and pushdown automata are the basis for many practical natural language processing systems. Algorithms for searching the combinatorial space of analyses of natural language utterances can be analyzed in terms of their computational complexity, and theoretically motivated approximations can sometimes be applied.

The study of computer systems is also relevant to natural language processing. Large datasets of unlabeled text can be processed more quickly by parallelization techniques like MapReduce (Dean and Ghemawat, 2008; Lin and Dyer, 2010); high-volume data sources such as social media can be summarized efficiently by approximate streaming and sketching techniques (Goyal et al., 2009). When deep neural networks are implemented in production systems, it is possible to eke out speed gains using techniques such as reduced-precision arithmetic (Wu et al., 2016). Many classical natural language processing algorithms are not naturally suited to graphics processing unit (GPU) parallelization, suggesting directions for further research at the intersection of natural language processing and computing hardware (Yi et al., 2011).

Speech Processing Natural language is often communicated in spoken form, and speech recognition is the task of converting an audio signal to text. From one perspective, this is a signal processing problem, which might be viewed as a preprocessing step before natural language processing can be applied. However, context plays a critical role in speech recognition by human listeners: knowledge of the surrounding words influences perception and helps to correct for noise (Miller et al., 1951). For this reason, speech recognition is often integrated with text analysis, particularly with statistical **language models**, which quantify the probability of a sequence of text (see chapter 6). Beyond speech recognition, the broader field of speech processing includes the study of speech-based dialogue systems, which are briefly discussed in chapter 19. Historically, speech processing has often been pursued in electrical engineering departments, while natural language processing has been the purview of computer scientists. For this reason, the extent of interaction between these two disciplines is less than it might otherwise be.

Ethics As machine learning and artificial intelligence become increasingly ubiquitous, it is crucial to understand how their benefits, costs, and risks are distributed across different kinds of people. Natural language processing raises some particularly salient issues around **ethics, fairness, and accountability**:

Access. Who is natural language processing designed to serve? For example, whose language is translated *from*, and whose language is translated *to*?

Bias. Does language technology learn to replicate social biases from text corpora, and does it reinforce these biases as seemingly objective computational conclusions?

Labor. Whose text and speech comprise the datasets that power natural language processing, and who performs the annotations? Are the benefits of this technology shared with all the people whose work makes it possible?

Privacy and internet freedom. What is the impact of large-scale text processing on the right to free and private communication? What is the potential role of natural language processing in regimes of censorship or surveillance?

This text lightly touches on issues related to fairness and bias in § 14.6.3 and § 18.1.1, but these issues are worthy of a book of their own. For more from within the field of computational linguistics, see the papers from the annual workshop on Ethics in Natural Language Processing (Hovy et al., 2017; Alfano et al., 2018). For an outside perspective on ethical issues relating to data science at large, see boyd and Crawford (2012).

Others Natural language processing plays a significant role in emerging interdisciplinary fields like **computational social science** and the **digital humanities**. Text classification (chapter 4), clustering (chapter 5), and information extraction (chapter 17) are particularly useful tools; another is **probabilistic topic models** (Blei, 2012), which are not covered in this text. **Information retrieval** (Manning et al., 2008) makes use of similar tools, and conversely, techniques such as latent semantic analysis (§ 14.3) have roots in information retrieval. **Text mining** is sometimes used to refer to the application of data mining techniques, especially classification and clustering, to text. While there is no clear distinction between text mining and natural language processing (nor between data mining and machine learning), text mining is typically less concerned with linguistic structure, and more interested in fast, scalable algorithms.

1.2 Three themes in natural language processing

Natural language processing covers a diverse range of tasks, methods, and linguistic phenomena. But despite the apparent incommensurability between, say, the summarization of scientific articles (§ 16.3.4) and the identification of suffix patterns in Spanish verbs (§ 9.1.4), some general themes emerge. The remainder of the introduction focuses on these themes, which will recur in various forms through the text. Each theme can be expressed as an opposition between two extreme viewpoints on how to process natural language. The methods discussed in the text can usually be placed somewhere on the continuum between these two extremes.

1.2.1 Learning and knowledge

A recurring topic of debate is the relative importance of machine learning and linguistic knowledge. On one extreme, advocates of "natural language processing from scratch" (Collobert et al., 2011) propose to use machine learning to train end-to-end systems that transmute raw text into any desired output structure: e.g., a summary, database, or translation. On the other extreme, the core work of natural language processing is sometimes taken to be transforming text into a stack of general-purpose linguistic structures: from subword units called **morphemes**, to word-level **parts-of-speech**, to tree-structured representations of grammar, and beyond, to logic-based representations of meaning. In theory, these general-purpose structures should then be able to support any desired application.

The end-to-end approach has been buoyed by recent results in computer vision and speech recognition, in which advances in machine learning have swept away expert-engineered representations based on the fundamentals of optics and phonology (Krizhevsky et al., 2012; Graves and Jaitly, 2014). But while machine learning is an element of nearly every contemporary approach to natural language processing, linguistic representations such as syntax trees have not yet gone the way of the visual edge detector or the auditory triphone. Linguists have argued for the existence of a "language faculty" in all human beings, which encodes a set of abstractions specially designed to facilitate the understanding and production of language. The argument for the existence of such a language faculty is based on the observation that children learn language faster and from fewer examples than would be possible if language was learned from experience alone.[4] From a practical standpoint, linguistic structure seems to be particularly important in scenarios where training data is limited.

There are a number of ways in which knowledge and learning can be combined in natural language processing. Many supervised learning systems make use of carefully engineered **features**, which transform the data into a representation that can facilitate

[4] *The Language Instinct* (Pinker, 2003) articulates these arguments in an engaging and popular style. For arguments against the innateness of language, see Elman et al. (1998).