

Book Review

Deep Learning for Natural Language Processing: A Gentle Introduction

Mihai Surdeanu and Marco Antonio Valenzuela-Escárcega
(University of Arizona)

Cambridge University Press, 2024, 344 pp; HARDBACK, 9781316515662; ebook,
9781009026222; doi: <https://doi.org/10.1017/9781009026222>

Reviewed by

Aline Paes

Federal Fluminense University, Brazil

The book *Deep Learning for Natural Language Processing: A Gentle Introduction* delivers precisely what its title promises: a pedagogical, conceptual, and practical exploration of deep learning techniques to address natural language processing (NLP) tasks. The book serves as an accessible guide for those new to machine learning and how its methods can be applied to NLP tasks. Each chapter thoughtfully introduces the necessary concepts while leading the reader step-by-step through illustrative examples and code. By including practical algorithms and code snippets, it provides a hands-on understanding of what occurs behind the scenes when deep learning models are applied to NLP problems. This approach makes the material more engaging and more accessible for learners to understand.

The text's structure is versatile enough to be adopted in undergraduate computer science courses and advanced courses for students from other disciplines, provided that they have some foundational understanding of programming. The emphasis on clarity and gradual introduction of complex ideas makes it an excellent choice for those who may initially find the subject difficult to understand. Despite the technical nature of the content, the authors manage to strike a balance between theory and practice, ensuring that readers from various backgrounds can follow along. In general, the book is a valuable resource for learners, providing a well-balanced mix of theory and practice.

If used as a course textbook, one drawback of the book is the absence of exercises in each chapter. Including exercises would strengthen its value as a teaching tool. A potential solution could be to incorporate tasks that involve modifying the provided code. This would allow readers to apply the concepts practically and deepen their understanding. Overall, the book serves as a foundational resource, offering an entry point into the world of machine learning for NLP, and provides a solid base for exploring more advanced topics later on.

The book is thoughtfully structured into 16 chapters and two appendices. It begins by setting the stage with an introduction to its scope and motivating the reader. Chapters 2 through 7 explore foundational machine learning (ML) concepts, emphasizing neural and deep networks, with text classification as the illustrative NLP task. The practical implementations utilize two widely adopted machine learning frameworks: PyTorch and Scikit-learn.

<https://doi.org/10.1162/COLLR.628>

Chapters 8 and 9 explore static word embeddings, grounded in the distributional hypothesis, offering clear explanations and practical examples. The book then transitions to addressing the sequential nature of text in Chapters 10 and 11, introducing recurrent neural networks and applying them to part-of-speech tagging tasks.

Chapters 12 through 15 focus on the cutting-edge advancements in NLP, presenting Transformers and encoder-decoder models, along with their implementations for machine translation using PyTorch and HuggingFace frameworks. The final chapter broadens the scope by covering additional NLP tasks, including named entity recognition, relation extraction, dependency parsing, and question answering. The appendices cover additional implementation details. In what follows, we provide more details about each chapter.

Chapter 1 introduces illustrative applications and basic concepts designed to engage and motivate the reader. It clearly outlines the scope of the book, detailing the NLP tasks and (ML) techniques it covers while clarifying the topics it does not address, such as ML formalisms beyond those based on deep learning (DL). The chapter also includes a valuable discussion on the limitations of DL, which helps readers understand the boundaries of methods grounded in DL. However, there are some drawbacks. The mention of a specific researcher and their former affiliation feels somewhat misplaced in a general discussion. Additionally, the provided definition of NLP could benefit from an expansion. Rather than framing NLP solely at the intersection of ML and AI, it is important to acknowledge that the field also draws from knowledge representation and other symbolic AI approaches. While ML is undeniably central to modern NLP, defining the field exclusively through ML overlooks these contributions. Moreover, the assertion that NLP depends primarily on large datasets is misleading. Many NLP tasks are successfully performed with small datasets, and the book would benefit from reflecting this broader perspective. Addressing these points would offer readers a more comprehensive and nuanced understanding of the field.

Chapter 2 centers on the perceptron, one of the simplest types of artificial neural networks, serving as the foundation for more advanced neural network architectures developed later. The chapter provides detailed code for each step of the algorithm, a practical approach consistently applied throughout the book. Presenting pseudocode before diving into implementation helps readers develop a more abstract understanding of the methods.

This chapter also includes a somewhat contentious discussion claiming that “ML is easy.” While some aspects of machine learning may seem intuitive, the underlying mathematical principles can be quite complex for those without the necessary background. Labeling machine learning as “easy” could frustrate or discourage readers who struggle. Acknowledging that some concepts are challenging but can be mastered with persistence would likely offer more encouragement.

The chapter’s historical overview of machine learning is informative, but readers should note that a deeper examination of subfields such as multi-label tasks, relational learning, graph-based learning, and unsupervised learning lies beyond the book’s scope and is available in the cited literature. While these topics are outside the book’s primary focus, mentioning them briefly would provide readers with a broader perspective on the field.

The explanation of overfitting requires refinement. Overfitting occurs when a model performs well on training data but fails to generalize to new, unseen data—often due to excessive complexity or insufficient and noisy data. Clarifying this concept would enhance the reader’s understanding. The mention of splitting data into training, validation, and test sets introduces a standard approach, but readers should also be made aware of more robust evaluation methods, such as cross-validation.

Lastly, the definition of “model” in the text feels unclear. It is ambiguous whether “model” refers to the learned parameters and the learning algorithm or to the parameters and the classifier. Providing a more precise explanation would prevent confusion and help readers better grasp the terminology used in machine learning.

Chapter 3 focuses on logistic regression, providing a comprehensive exploration of this widely used technique and highlighting how it differs from the perceptron steps introduced in the previous chapter. The chapter begins by outlining the core principles of logistic regression, emphasizing its role as a probabilistic model in contrast to the perceptron’s deterministic approach.

The book thoroughly explains the learning algorithm for logistic regression, detailing the update rule that adjusts the model’s weights during training to minimize the loss function (commonly the cross-entropy loss). It includes both the mathematical derivation and the pseudocode of this rule, making it accessible for readers with varying levels of mathematical proficiency.

In addition to the foundational concepts, the chapter addresses logistic regression’s formulation for both binary and multiclass classification problems. For binary classification, it explores the sigmoid activation function and its relationship to the likelihood of class membership. For multiclass problems, it introduces the softmax function and explains how logistic regression can extend to handle multiple categories effectively.

The chapter also covers evaluation metrics for assessing logistic regression models, including confusion matrix, precision, recall, F1 score, together with macro and micro averaging. Finally, the chapter discusses the limitations of logistic regression, such as its reliance on a linear decision boundary and potential struggles with complex, non-linear relationships in data.

Chapter 4 provides a detailed, step-by-step guide for implementing a text classification task using perceptron and logistic regression, both of which were introduced in more general terms in the previous chapters. Text classification is addressed as both a binary and a multiclass task. The chapter also introduces the bag-of-words model, a foundational technique for converting text into numerical features required by machine learning models. Several implementation snippets are included, enhancing the reader’s practical understanding and ability to apply these concepts. The results of the implemented task are also presented using the precision, recall, and F1 metrics introduced in the previous chapter.

Readers should take note of specific implementation terms introduced in this chapter, such as “unsqueeze.” For a deeper understanding, they are encouraged to consult Appendix 1 or refer to the PyTorch documentation (<https://pytorch.org/docs/stable/generated/torch.unsqueeze.html>).

As the chapter looks into text classification, the reader should be aware of the importance of preprocessing text by breaking it down into smaller components. This is typically achieved through tokenization, a key step in NLP. Tokenization involves dividing text into smaller units called tokens, which can range from individual characters to words or subwords. In some cases, tokenization may also involve splitting text into sentences.

Chapter 5 focuses on the motivation, concepts, and architecture of feed-forward neural networks (FFNN), specifically implemented as multi-layer perceptrons (MLP). It thoroughly explains and derives the backpropagation algorithm, the widely adopted algorithm for adjusting neural network weights. The chapter also discusses the significance of multiple layers in capturing complex patterns and the optimization strategies used to learn weights effectively.

Chapter 6 acts as a practical extension of Chapter 5, detailing variations in MLP architecture and training best practices. It introduces mini-batching, a common technique in deep learning that involves splitting the training dataset into smaller, manageable groups of samples called mini-batches. These mini-batches are processed sequentially during training, improving computational efficiency. The chapter also discusses stochastic gradient descent, an optimization algorithm designed to make gradient descent scalable and efficient for large datasets. Other optimization methods, such as Momentum, Adagrad, and Adadelta, are briefly covered, with an emphasis on their specific purposes, such as smoothing updates or adapting learning rates. In addition, the chapter explores various strategies aimed at improving training efficiency and generalization, including alternative activation functions to the logistic function explored in previous chapters, like ReLU; loss (cost) functions, including an extended explanation of the cross-entropy introduced in Chapter 3 metrics for evaluating model performance and guiding optimization; regularization techniques to reduce overfitting; dropout to enhance generalization by randomly disabling neurons during training, and parameter initialization strategies.

Building on the framework established in Chapter 4 for addressing text classification and the neural network concepts introduced in Chapters 5 and 6, Chapter 7 provides a step-by-step guide for implementing text classification using FFNN. The chapter includes pseudocode to help readers understand the implementation at an abstract level, followed by detailed explanations of the code itself. Additionally, it presents the results of the implemented text classification task, evaluated using metrics introduced earlier, such as precision, recall, and F1 score.

Chapter 8 introduces the concept of representing words as vectors, exploring key ideas such as the distributional hypothesis, matrix decomposition, topic modeling, and the algorithms implemented in Word2Vec. Word2Vec, a widely used software based on neural networks, generates dense vector representations of words (word embeddings) that align with the distributional hypothesis: words appearing in similar contexts have similar meanings. The chapter highlights foundational ideas but could benefit from greater clarity in certain areas. For example, the explanation of the “curse of dimensionality” does not fully emphasize its core challenge: High dimensionality results in sparse data, which complicates the learning process. Additionally, Word2Vec is referred to as an “algorithm”, but it should be more accurately described as a software package that implements two distinct algorithms: Skip-gram and continuous bag-of-words (CBOW). Readers should recognize that Skip-gram operates on pairs of words, predicting a context word given a target word (or vice versa), rather than processing entire contexts at once. This distinction is crucial to avoid misconceptions about how these algorithms function.

Chapter 9 builds on the concepts introduced in Chapter 8, providing a practical implementation of text classification using word embeddings obtained from Word2Vec and Gensim, a widely used framework for NLP tasks.

The chapter offers a detailed, step-by-step explanation of the code, guiding readers through the process of integrating pre-trained word embeddings into a text classification pipeline. This includes loading embeddings from Word2Vec, preparing data for classification, and leveraging Gensim’s functionality to manage and manipulate word vectors effectively. Once again, this highlights the specific nature of the book, on bridging the gap between theory and practical application, making it easier for readers to apply these techniques to their own NLP projects.

Chapter 10 introduces more advanced neural network architectures for handling NLP tasks, specifically recurrent neural networks (RNNs). These networks possess the

inherent ability to consider the order of words in a text, making them well-suited for sequential data processing.

The chapter covers vanilla RNNs as well as their improved variant, long short-term memory (LSTM) networks. LSTMs are designed to address the vanishing and exploding gradient problems that often hinder the training of standard RNNs. The explanation is enriched with practical insights, using part-of-speech tagging as an illustrative task to demonstrate the nuances of recurrent neural networks and their application in real-world scenarios.

For readers seeking a deeper understanding of the vanishing-gradient issue, the chapter provides references to additional literature, offering opportunities for further exploration.

Following the recurrent setting of the book, Chapter 11 explains the step-by-step code for handling parts of speech with recurrent neural networks using PyTorch. An example illustrating padding and packing transformations (as described on page 172) would greatly benefit the reader's understanding of these important preprocessing steps. Padding involves adding extra elements (usually zeros) to sequences so that all sequences in a batch have the same length. This is crucial for processing variable-length sequences, as many deep learning frameworks require inputs to have uniform dimensions. Packing optimizes processing by excluding padded elements when working with padded sequences.

Chapter 12 introduces contextualized word embeddings, a transformative advancement in NLP that captures the meaning of words based on their surrounding context. This approach addresses the limitations of static embeddings introduced in Chapter 8, offering more nuanced representations of language. The chapter presents and discusses the transformer architecture, a cornerstone of modern NLP models, breaking it down into its key components and concepts of the encoder-component, mainly responsible for processing and transforming input text into a compact, meaningful, and contextualized representation.

The chapter discusses that at the heart of the Transformer encoder is the self-attention mechanism, which allows the model to focus on relevant parts of a sequence, capturing relationships between words regardless of their position in the text. Complementing this is positional encoding, which ensures that the model accounts for the order of words, compensating for the lack of inherent sequential processing in the architecture. These features are further enhanced by feed-forward networks, which refine and transform the representations generated during processing.

The chapter also explores the learning paradigms that have made these architectures so impactful. Pre-training is introduced as a foundational step, where models are trained on massive datasets using unsupervised objectives like masked language modeling to learn general linguistic patterns. This is followed by fine-tuning, a process that adapts these pre-trained models to specific tasks, such as sentiment analysis or named entity recognition, using labeled data tailored to the task at hand. Additionally, the chapter examines the tokenization of text into subwords, a technique that reduces the occurrence of out-of-vocabulary words. While this discussion is insightful, it could benefit from a deeper exploration of specific algorithms used for subword tokenization.

The reader should note that Figure 12.1 offers an oversimplified view of Transformer operations on word inputs. While this may be intentional for pedagogical reasons, a more detailed explanation or note could prevent readers from assuming the process is simpler than it is in practice.

Chapters 13–15 serve as practical complements to the theoretical foundation laid in Chapter 12, providing hands-on insights into the implementation and application of transformers and related architectures.

Chapter 13 introduces the implementation of Transformer models using the HuggingFace Hub, a widely adopted platform for deploying pre-trained models and experimenting with state-of-the-art NLP architectures. The chapter shows how to integrate these pre-trained models into practical workflows, highlighting their versatility and ease of use. Chapter 14 shifts focus to the more general idea of encoder-decoder architectures, motivated by its application to the machine translation task, where the encoder processes the input text (e.g., a sentence in one language) into a contextualized representation, and the decoder generates output text (e.g., the translation in another language). The chapter provides a clear explanation of how this paradigm works and its adaptability to other sequence-to-sequence tasks.

Building on this foundation, Chapter 15 takes a more practical look into the implementation of the encoder-decoder architecture, moving beyond the pre-packaged solutions available in HuggingFace. This chapter provides a more detailed, ground-up implementation, offering readers a deeper understanding of how these architectures function under the hood. It emphasizes key aspects like custom model design, training pipelines, and evaluation, empowering readers to modify and adapt these architectures for specialized tasks.

Chapter 16 concludes the book by exploring a variety of advanced NLP tasks and the deep learning architectures used to solve them. It introduces several important applications that were not extensively covered earlier, such as question answering, named entity recognition (NER), dependency parsing, and relation extraction. These tasks highlight the versatility of deep learning and its ability to address diverse challenges in natural language processing.

Question answering systems are designed to respond to questions posed in natural language by extracting relevant information from structured databases or unstructured text. NER focuses on identifying and classifying entities within text into categories such as people, organizations, locations, dates, or numerical values. Dependency parsing examines the grammatical structure of sentences by identifying relationships between words, such as subject-verb and object-verb dependencies. This task is fundamental for syntactic analysis and is widely used in applications like machine translation and text-to-speech systems. Relation extraction identifies and categorizes semantic relationships between entities within a text.

Although these tasks could have been introduced earlier in the book, their placement in the final chapter is purposeful. By presenting them after establishing a strong foundation in NLP and deep learning, the book ensures readers are well-prepared to understand and apply these advanced concepts. This approach not only reinforces the material covered in earlier chapters but also provides a comprehensive view of the field, underscoring how deep learning can address complex language tasks.

To sum up, the book provides a comprehensive journey through the foundations and advancements in natural language processing using deep learning. Starting with fundamental concepts like machine learning basics, text classification, and static embeddings, it gradually builds up to more sophisticated topics such as contextualized embeddings, Transformer architectures, and their practical applications. The clear explanations of theoretical concepts, paired with step-by-step implementation guides, ensure readers gain both a solid understanding and hands-on experience.

Aline Paes is an Associate Professor at the Institute of Computing at the Federal Fluminense University (UFF). She earned her master's and doctorate in Systems and Computing Engineering, specializing in Artificial Intelligence, from COPPE-Systems at UFRJ. She also spent a year as a visiting doctoral student at Imperial College London, UK. She has a Young Scientist of Rio de Janeiro State fellowship from FAPERJ and a productivity scholarship from CNPq. Aline Paes's

research in Artificial Intelligence spans several areas: machine learning integrated with neural, statistical, and logical techniques; natural language representation learning; model adaptation; transfer learning; explainable AI; and AI for positive social impact. She serves on the editorial boards of the *Machine Learning Journal*, the *Ibero-American Journal of Artificial Intelligence*, and the *Journal of the Brazilian Computer Society*. She was recently awarded the inaugural FAPERJ research project for Young Women Scientists. Since 2020, she has been a member of the Brazilian Women in NLP group (Brasileiras em PLN). In 2023, she was a visiting professor in the NLP Group at the University of Sheffield, supported by a scholarship from CAPES. Paes's e-mail address is alinepaes@ic.uff.br.