

Thushan Ganegedara 著

---

# TensorFlow 自然语言处理

(影印版)

Natural Language Processing with TensorFlow

---

 东南大学出版社  
SOUTHEAST UNIVERSITY PRESS



TensorFlow 自然语言处理(影印版)  
Natural Language Processing  
with TensorFlow

Thushan Ganegedara 著

南京 东南大学出版社

## 图书在版编目(CIP)数据

TensorFlow 自然语言处理:英文/(澳)苏尚·甘吉达拉(Thushan Ganegedara)著. —影印本. —南京:东南大学出版社,2019.3

书名原文:Natural Language Processing with TensorFlow  
ISBN 978-7-5641-8289-2

I. ①T… II. ①苏… III. ①人工智能—算法—英文  
IV. ①TP18

中国版本图书馆 CIP 数据核字(2019)第 025338 号  
图字:10-2018-491 号

© 2018 by PACKT Publishing Ltd.

Reprint of the English Edition, jointly published by PACKT Publishing Ltd and Southeast University Press, 2019.  
Authorized reprint of the original English edition, 2018 PACKT Publishing Ltd, the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 PACKT Publishing Ltd 出版 2018。

英文影印版由东南大学出版社出版 2019。此影印版的出版和销售得到出版权和销售权的所有者——PACKT Publishing Ltd 的许可。

版权所有,未得书面许可,本书的任何部分和全部不得以任何形式复制。

## TensorFlow 自然语言处理(影印版)

出版发行:东南大学出版社

地 址:南京四牌楼 2 号 邮编:210096

出 版 人:江建中

网 址:<http://www.seupress.com>

电子邮件:[press@seupress.com](mailto:press@seupress.com)

印 刷:常州市武进第三印刷有限公司

开 本:787 毫米×980 毫米 16 开本

印 张:29.5

字 数:578 千字

版 次:2019 年 3 月第 1 版

印 次:2019 年 3 月第 1 次印刷

书 号:ISBN 978-7-5641-8289-2

定 价:106.00 元

本社图书若有印装质量问题,请直接与营销部联系。电话(传真):025-83791830



mapt.io

Mapt is an online digital library that gives you full access to over 5,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

## Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Learn better with Skill Plans built especially for you
- Get a free eBook or video every month
- Mapt is fully searchable
- Copy and paste, print, and bookmark content

## PacktPub.com

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

# Contributors

## About the author

**Thushan Ganegedara** is currently a third year Ph.D. student at the University of Sydney, Australia. He is specializing in machine learning and has a liking for deep learning. He lives dangerously and runs algorithms on untested data. He also works as the chief data scientist for AssessThreat, an Australian start-up. He got his BSc. (Hons) from the University of Moratuwa, Sri Lanka. He frequently writes technical articles and tutorials about machine learning. Additionally, he also strives for a healthy lifestyle by including swimming in his daily schedule.

---

I would like to thank my parents, my siblings, and my wife for the faith they had in me and the support they have given, also all my teachers and my Ph.D advisor for the guidance he provided me with.

---

## About the reviewers

**Motaz Saad** holds a Ph.D. in computer science from the University of Lorraine. He loves data and he likes to play with it. He has over 10 years, professional experience in NLP, computational linguistics, data science, and machine learning. He currently works as an assistant professor at the faculty of information technology, IUG.

**Dr Joseph O'Connor** is a data scientist with a deep passion for deep learning. His company, Deep Learn Analytics, a UK-based data science consultancy, works with businesses to develop machine learning applications and infrastructure from concept to deployment. He was awarded a Ph.D. from University College London for his work analyzing data on the MINOS high-energy physics experiment. Since then, he has developed ML products for a number of companies in the private sector, specializing in NLP and time series forecasting. You can find him at <http://deeplearnanalytics.com/>.

## Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit [authors.packtpub.com](http://authors.packtpub.com) and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

# Preface

In the digital information age that we live in, the amount of data has grown exponentially, and it is growing at an unprecedented rate as we read this. Most of this data is language-related data (textual or verbal), such as emails, social media posts, phone calls, and web articles. Natural Language Processing (NLP) leverages this data efficiently to help humans in their businesses or day-to-day tasks. NLP has already revolutionized the way we use data to improve both businesses and our lives, and will continue to do so in the future.

One of the most ubiquitous use cases of NLP is Virtual Assistants (VAs), such as Apple's Siri, Google Assistant, and Amazon Alexa. Whenever you ask your VA for "the cheapest rates for hotels in Switzerland," a complex series of NLP tasks are triggered. First, your VA needs to understand (parse) your request (for example, learn that it needs to retrieve hotel rates, not the dog parks). Another decision the VA needs to make is "what is cheap?". Next, the VA needs to rank the cities in Switzerland (perhaps based on your past traveling history). Then, the VA might crawl websites such as Booking.com and Agoda.com to fetch the hotel rates in Switzerland and rank them by analyzing both the rates and reviews for each hotel. As you can see, the results you see in a few seconds are a result of a very intricate series of complex NLP tasks.

So, what makes such NLP tasks so versatile and accurate for our everyday tasks? The underpinning elements are "deep learning" algorithms. Deep learning algorithms are essentially complex neural networks that can map raw data to a desired output without requiring any sort of task-specific feature engineering. This means that you can provide a hotel review of a customer and the algorithm can answer the question "How positive is the customer about this hotel?", directly. Also, deep learning has already reached, and even exceeded, human-level performance in a variety of NLP tasks (for example, speech recognition and machine translation).

By reading this book, you will learn how to solve many interesting NLP problems using deep learning. So, if you want to be an influencer who changes the world, studying NLP is critical. These tasks range from learning the semantics of words, to generating fresh new stories, to performing language translation just by looking at bilingual sentence pairs. All of the technical chapters are accompanied by exercises, including step-by-step guidance for readers to implement these systems. For all of the exercises in the book, we will be using Python with TensorFlow – a popular distributed computation library that makes implementing deep neural networks very convenient.

## Who this book is for

This book is for aspiring beginners who are seeking to transform the world by leveraging linguistic data. This book will provide you with a solid practical foundation for solving NLP tasks. In this book, we will cover various aspects of NLP, focusing more on the practical implementation than the theoretical foundation. Having sound practical knowledge of solving various NLP tasks will help you to have a smoother transition when learning the more advanced theoretical aspects of these methods. In addition, a solid practical understanding will help when performing more domain-specific tuning of your algorithms, to get the most out of a particular domain.

## What this book covers

*Chapter 1, Introduction to Natural Language Processing*, embarks us on our journey with a gentle introduction to NLP. In this chapter, we will first look at the reasons we need NLP. Next, we will discuss some of the common subtasks found in NLP. Thereafter, we will discuss the two main eras of NLP – the traditional era and the deep learning era. We will gain an understanding of the characteristics of the traditional era by working through how a language modeling task might have been solved with traditional algorithms. Then, we will discuss the deep learning era, where deep learning algorithms are heavily utilized for NLP. We will also discuss the main families of deep learning algorithms. We will then discuss the fundamentals of one of the most basic deep learning algorithms – a fully connected neural network. We will conclude the chapter with a road map that provides a brief introduction to the coming chapters.

*Chapter 2, Understanding TensorFlow*, introduces you to the Python TensorFlow library – the primary platform we will implement our solutions on. We will start by writing code to perform a simple calculation in TensorFlow. We will then discuss how things are executed, starting from running the code to getting results. Thereby, we will understand the underlying components of TensorFlow in detail. We will further strengthen our understanding of TensorFlow with a colorful analogy of a restaurant and see how orders are fulfilled. Later, we will discuss more technical details of TensorFlow, such as the data structures and operations (mostly related to neural networks) defined in TensorFlow. Finally, we will implement a fully connected neural network to recognize handwritten digits. This will help us to understand how an end-to-end solution might be implemented with TensorFlow.

*Chapter 3, Word2vec – Learning Word Embeddings*, begins by discussing how to solve NLP tasks with TensorFlow. In this chapter, we will see how neural networks can be used to learn word vectors or word representations. Word vectors are also known as word embeddings. Word vectors are numerical representations of words that have similar values for similar words and different values for different words. First, we will discuss several traditional approaches to achieving this, which include using a large human-built knowledge base known as WordNet. Then, we will discuss the modern neural network-based approach known as Word2vec, which learns word vectors without any human intervention. We will first understand the mechanics of Word2vec by working through a hands-on example. Then, we will discuss two algorithmic variants for achieving this – the skip-gram and continuous bag-of-words (CBOW) model. We will discuss the conceptual details of the algorithms, as well as how to implement them in TensorFlow.

*Chapter 4, Advance Word2vec*, takes us on to more advanced topics related to word vectors. First, we will compare skip-gram and CBOW to see whether a winner exists. Next, we will discuss several improvements that can be used to improve the performance of the Word2vec algorithms. Then, we will discuss a more recent and powerful word embedding learning algorithm – the GloVe (global vectors) algorithm. Finally, we will look at word vectors in action, in a document classification task. In that exercise, we will see that word vectors are powerful enough to represent the topic (for example, entertainment and sport) that the document belongs to.

*Chapter 5, Sentence Classification with Convolutional Neural Networks*, discusses convolution neural networks (CNN) – a family of neural networks that excels at processing spatial data such as images or sentences. First, we will develop a solid high-level understanding of CNNs by discussing how they process data and what sort of operations are involved. Next, we will dive deep into each of the operations involved in the computations of a CNN to understand the underpinning mathematics of a CNN. Finally, we will walk through two exercises. First, we will classify hand written digit images with a CNN. We will see that CNNs are capable of reaching a very high accuracy quickly for this task. Next, we will explore how CNNs can be used to classify sentences. Particularly, we will ask a CNN to predict whether a sentence is about an object, person, location, and so on.

*Chapter 6, Recurrent Neural Networks*, is about a powerful family of neural networks that can model sequences of data, known as recurrent neural networks (RNNs). We will first discuss the mathematics behind the RNNs and the update rules that are used to update the RNNs over time during learning. Then, we will discuss different variants of RNNs and their applications (for example, one-to-one RNNs and one-to-many RNNs). Finally, we will go through an exercise where RNNs are used for a text generation task. In this, we will train the RNN on folk stories and ask the RNN to produce a new story. We will see that RNNs are poor at persisting long-term memory. Finally, we will discuss a more advanced variant of RNNs, which we will call RNN-CF, which is able to persist memory for longer.

*Chapter 7, Long Short-Term Memory Networks*, allows us to explore more powerful techniques that are able to remember for a longer period of time, having found out that RNNs are poor at retaining long-term memory. We will discuss one such technique in this chapter – Long Short-Term Memory Networks (LSTMs). LSTMs are more powerful and have been shown to outperform other sequential models in many time-series tasks. We will first investigate the underlying mathematics and update the rules of the LSTM, along with a colorful example that illustrates why each computation matters. Then, we will look at how LSTMs can persist memory for longer. Next, we will discuss how we can improve LSTMs prediction capabilities further. Finally, we will discuss several variants of LSTMs that have a more complex structure (LSTMs with peephole connections), as well as a method that tries to simplify the LSTMs gated recurrent units (GRUs).

*Chapter 8, Applications of LSTM – Generating Text*, extensively evaluates how LSTMs perform in a text generation task. We will qualitatively and quantitatively measure how good the text generated by LSTMs is. We will also conduct comparisons between LSTMs, LSTMs with peephole connections, and GRUs. Finally, we will see how we can bring word embeddings into the model to improve the text generated by LSTMs.

*Chapter 9, Applications of LSTM – Image Caption Generation*, moves us on to multimodal data (that is, images and text) after coping with textual data. In this chapter, we will investigate how we can automatically generate descriptions for a given image. This involves combining a feed-forward model (that is, a CNN) with a word embedding layer and a sequential model (that is, an LSTM) in a way that forms an end-to-end machine learning pipeline.

*Chapter 10, Sequence to Sequence Learning – Neural Machine Translation*, is about the implementing neural machine translation (NMT) model. Machine translation is where we translate a sentence/phrase from a source language into a target language. We will first briefly discuss what machine translation is. This will be followed by a section about the history of machine translation. Then, we will discuss the architecture of modern neural machine translation models in detail, including the training and inference procedures. Next, we will look at how to implement an NMT system from scratch. Finally, we will explore ways to improve standard NMT systems.

*Chapter 11, Current Trends and Future of Natural Language Processing*, the final chapter, focuses on the current and future trends of NLP. We will discuss the latest discoveries related to the systems and tasks we discussed in the previous chapters. This chapter will cover most of the exciting novel innovations, as well as giving you in-depth intuition to implement some of the technologies.

*Appendix, Mathematical Foundations and Advanced TensorFlow*, will introduce the reader to various mathematical data structures (for example, matrices) and operations (for example, matrix inverse). We will also discuss several important concepts in probability. We will then introduce Keras – a high-level library that uses TensorFlow underneath. Keras makes the implementing of neural networks simpler by hiding some of the details in TensorFlow, which some might find challenging. Concretely, we will see how we can implement a CNN with Keras, to get a feel of how to use Keras. Next, we will discuss how we can use the seq2seq library in TensorFlow to implement a neural machine translation system with much less code that we used in Chapter 11, Current Trends and the Future of Natural Language Processing. Finally, we will walk you through a guide aimed at teaching to use the TensorBoard to visualize word embeddings. TensorBoard is a handy visualization tool that is shipped with TensorFlow. This can be used to visualize and monitor various variables in your TensorFlow client.

# To get the most out of this book

To get the most out of this book, we assume the following from the reader:

- A solid will and an ambition to learn the modern ways of NLP
- Familiarity with basic Python syntax and data structures (for example, lists and dictionaries)
- A good understanding of basic mathematics (for example, matrix/vector multiplication)
- (Optional) Advance mathematics knowledge (for example, derivative calculation) to understand a handful of subsections that cover the details of how certain learning models overcome potential practical issues faced during training
- (Optional) Read research papers to refer to advances/details in systems, beyond what the book covers

## Download the example code files

You can download the example code files for this book from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files emailed directly to you.

You can download the code files by following these steps:

1. Log in or register at <http://www.packtpub.com>.
2. Select the **SUPPORT** tab.
3. Click on **Code Downloads & Errata**.
4. Enter the name of the book in the **Search** box and follow the on-screen instructions.

Once the file is downloaded, please make sure that you unzip or extract the folder using the latest version of one of these:

- WinRAR / 7-Zip for Windows
- Zipeg / iZip / UnRarX for macOS
- 7-Zip / PeaZip for Linux

The code bundle for the book is also hosted on GitHub at <https://github.com/PacktPublishing/Natural-Language-Processing-with-TensorFlow>. We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing/>. Check them out!

## Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: [https://www.packtpub.com/sites/default/files/downloads/NaturalLanguageProcessingwithTensorFlow\\_ColorImages.pdf](https://www.packtpub.com/sites/default/files/downloads/NaturalLanguageProcessingwithTensorFlow_ColorImages.pdf).

## Conventions used

There are a number of text conventions used throughout this book.

**CodeInText**: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. For example; "Mount the downloaded `WebStorm-10*.dmg` disk image file as another disk in your system."

A block of code is set as follows:

```
graph = tf.Graph() # Creates a graph
session = tf.InteractiveSession(graph=graph) # Creates a session
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
graph = tf.Graph() # Creates a graph
session = tf.InteractiveSession(graph=graph) # Creates a session
```

Any command-line input or output is written as follows:

```
conda --version
```

**Bold**: Indicates a new term, an important word, or words that you see on the screen, for example, in menus or dialog boxes, also appear in the text like this. For example: "Select **System info** from the **Administration** panel."

*References*: In *Chapter 11, Current Trends and the Future of Natural Language Processing*, in-text references include a bracketed number (for example, [1]) that correlates with the numbering in the *References* section at the end of the chapter.



Warnings or important notes appear like this.



Tips and tricks appear like this.

## Get in touch

Feedback from our readers is always welcome.

**General feedback:** Email [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book's title in the subject of your message. If you have questions about any aspect of this book, please email us at [questions@packtpub.com](mailto:questions@packtpub.com).

**Errata:** Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book we would be grateful if you would report this to us. Please visit, <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the Errata Submission Form link, and entering the details.

**Piracy:** If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the material.

**If you are interested in becoming an author:** If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit <http://authors.packtpub.com>.

## Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit [packtpub.com](http://packtpub.com).

# Table of Contents

<b>Preface</b>	<b>xi</b>
<b>Chapter 1: Introduction to Natural Language Processing</b>	<b>1</b>
<b>What is Natural Language Processing?</b>	<b>1</b>
<b>Tasks of Natural Language Processing</b>	<b>2</b>
<b>The traditional approach to Natural Language Processing</b>	<b>5</b>
Understanding the traditional approach	5
Example – generating football game summaries	6
Drawbacks of the traditional approach	10
<b>The deep learning approach to Natural Language Processing</b>	<b>10</b>
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
<b>The roadmap – beyond this chapter</b>	<b>16</b>
<b>Introduction to the technical tools</b>	<b>21</b>
Description of the tools	21
Installing Python and scikit-learn	22
Installing Jupyter Notebook	22
Installing TensorFlow	23
<b>Summary</b>	<b>24</b>
<b>Chapter 2: Understanding TensorFlow</b>	<b>27</b>
<b>What is TensorFlow?</b>	<b>28</b>
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
<b>Inputs, variables, outputs, and operations</b>	<b>36</b>
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
<b>Reusing variables with scoping</b>	<b>57</b>
<b>Implementing our first neural network</b>	<b>59</b>
Preparing the data	60
Defining the TensorFlow graph	61
Running the neural network	63
<b>Summary</b>	<b>65</b>
<b>Chapter 3: Word2vec – Learning Word Embeddings</b>	<b>67</b>
<b>What is a word representation or meaning?</b>	<b>69</b>
<b>Classical approaches to learning word representation</b>	<b>69</b>
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
<b>Word2vec – a neural network-based approach to learning word representation</b>	<b>77</b>
Exercise: is queen = king – he + she?	78
Designing a loss function for learning word embeddings	82
<b>The skip-gram algorithm</b>	<b>83</b>
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
<b>The Continuous Bag-of-Words algorithm</b>	<b>98</b>
Implementing CBOW in TensorFlow	99
<b>Summary</b>	<b>100</b>
<b>Chapter 4: Advanced Word2vec</b>	<b>103</b>
<b>The original skip-gram algorithm</b>	<b>104</b>
Implementing the original skip-gram algorithm	105

Comparing the original skip-gram with the improved skip-gram	107
<b>Comparing skip-gram with CBOW</b>	<b>107</b>
Performance comparison	108
Which is the winner, skip-gram or CBOW?	112
<b>Extensions to the word embeddings algorithms</b>	<b>114</b>
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
<b>More recent algorithms extending skip-gram and CBOW</b>	<b>119</b>
A limitation of the skip-gram algorithm	119
The structured skip-gram algorithm	120
The loss function	120
The continuous window model	122
<b>GloVe – Global Vectors representation</b>	<b>123</b>
Understanding GloVe	123
Implementing GloVe	125
<b>Document classification with Word2vec</b>	<b>126</b>
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
<b>Summary</b>	<b>134</b>
<b>Chapter 5: Sentence Classification with Convolutional Neural Networks</b>	<b>135</b>
<b>Introducing Convolution Neural Networks</b>	<b>136</b>
CNN fundamentals	136
The power of Convolution Neural Networks	139
<b>Understanding Convolution Neural Networks</b>	<b>139</b>
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