

Table of contents

1. [Title Page](#)
2. [Copyright and Credits](#)
 1. [Natural Language Processing with Python Quick Start Guide](#)
3. [About Packt](#)
 1. [Why subscribe?](#)
 2. [Packt.com](#)
4. [Contributors](#)
 1. [About the author](#)
 2. [About the reviewer](#)
 3. [Packt is searching for authors like you](#)
5. [Preface](#)
 1. [Who this book is for](#)
 2. [What this book covers](#)
 3. [To get the most out of this book](#)
 1. [Download the example code files](#)
 2. [Download the color images](#)
 3. [Conventions used](#)
 4. [Get in touch](#)
 1. [Reviews](#)
6. [Getting Started with Text Classification](#)
 1. [What is NLP?](#)
 1. [Why learn about NLP?](#)
 1. [You have a problem in mind](#)
 2. [Technical achievement](#)
 3. [Do something new](#)
 4. [Is this book for you?](#)
 2. [NLP workflow template](#)
 1. [Understanding the problem](#)
 1. [Understanding and preparing the data](#)
 2. [Quick wins #x2013;#xA0;proof of concept](#)
 3. [Iterating and improving](#)
 1. [Algorithms](#)
 2. [Pre-processing](#)
 4. [Evaluation and deployment](#)
 1. [Evaluation](#)
 2. [Deployment](#)
 3. [Example #xA0;#x2013;#xA0;text classification workflow](#)
 1. [Launchpad #xA0;#x2013;#xA0;programming environment setup](#)
 1. [Text classification in 30 lines of code](#)
 1. [Getting the data](#)
 2. [Text to numbers](#)
 3. [Machine learning](#)
 4. [Summary](#)
 7. [Tidying your Text](#)

1. [Bread and butter – most common tasks](#)
 1. [Loading the data](#)
 2. [Exploring the loaded data](#)
2. [Tokenization](#)
 1. [Intuitive – split by whitespace](#)
 2. [The hack – splitting by word extraction](#)
 1. [Introducing Regexes](#)
 3. [spaCy for tokenization](#)
 1. [How does the spaCy tokenizer work?](#)
 2. [Sentence tokenization](#)
 4. [Stop words removal and case change](#)
3. [Stemming and lemmatization](#)
 1. [spaCy for lemmatization](#)
 1. [-PRON-](#)
 2. [Case-insensitive](#)
 3. [Conversion – meeting to meet](#)
4. [spaCy compared with NLTK and CoreNLP](#)
5. [Correcting spelling](#)
 1. [FuzzyWuzzy](#)
 2. [Jellyfish](#)
 3. [Phonetic word similarity](#)
 1. [What is a phonetic encoding?](#)
 2. [Runtime complexity](#)
6. [Cleaning a corpus with FlashText](#)
7. [Summary](#)
8. [Leveraging Linguistics](#)
 1. [Linguistics and NLP](#)
 1. [Getting started](#)
 2. [Introducing textacy](#)
 3. [Redacting names with named entity recognition](#)
 1. [Entity types](#)
 4. [Automatic question generation](#)
 1. [Part-of-speech tagging](#)
 2. [Creating a ruleset](#)
 5. [Question and answer generation using dependency parsing](#)
 1. [Visualizing the relationship](#)
 2. [Introducing textacy](#)
 3. [Leveling up – question and answer](#)
 6. [Putting it together and the end](#)
 2. [Summary](#)
9. [Text Representations - Words to Numbers](#)
 1. [Vectorizing a specific dataset](#)
 2. [Word representations](#)
 1. [How do we use pre-trained embeddings?](#)
 2. [KeyedVectors API](#)
 1. [What is missing in both word2vec and GloVe?](#)

3. [How do we handle Out Of Vocabulary words?](#)
 1. [Getting the dataset](#)
 4. [Training fastText embeddings](#)
 5. [Training word2vec embeddings](#)
 6. [fastText versus word2vec](#)
 3. [Document embedding](#)
 1. [Understanding the doc2vec API](#)
 1. [Negative sampling](#)
 2. [Hierarchical softmax](#)
 2. [Data exploration and model evaluation](#)
 4. [Summary](#)
10. [Modern Methods for Classification](#)
 1. [Machine learning for text](#)
 1. [Sentiment analysis as text classification](#)
 1. [Simple classifiers](#)
 2. [Optimizing simple classifiers](#)
 3. [Ensemble methods](#)
 2. [Getting the data](#)
 1. [Reading data](#)
 3. [Simple classifiers](#)
 1. [Logistic regression](#)
 1. [Removing stop words](#)
 2. [Increasing ngram range](#)
 2. [Multinomial Naive Bayes](#)
 1. [Adding TF-IDF](#)
 2. [Removing stop words](#)
 3. [Changing fit prior to false](#)
 3. [Support vector machines](#)
 4. [Decision trees](#)
 5. [Random forest classifier](#)
 6. [Extra trees classifier](#)
 4. [Optimizing our classifiers](#)
 1. [Parameter tuning using RandomizedSearch](#)
 1. [GridSearch](#)
 5. [Ensembling models](#)
 1. [Voting ensembles](#); Simple majority (aka hard voting)
 2. [Voting ensembles](#); soft voting
 3. [Weighted classifiers](#)
 4. [Removing correlated classifiers](#)
 2. [Summary](#)
11. [Deep Learning for NLP](#)
 1. [What is deep learning?](#)
 1. [Differences between modern machine learning methods](#)
 2. [Understanding deep learning](#)
 1. [Puzzle pieces](#)
 1. [Model](#)

2. [Loss function](#)
 3. [Optimizer](#)
 3. [Putting it all together – the training loop](#)
 4. [Kaggle – text categorization challenge](#)
 1. [Getting the data](#)
 2. [Exploring the data](#)
 1. [Multiple target dataset](#)
 2. [Why PyTorch?](#)
 3. [PyTorch and torchtext](#)
 3. [Data loaders with torchtext](#)
 4. [Conventions and style](#)
 1. [Knowing the field](#)
 5. [Exploring the dataset objects](#)
 6. [Iterators](#)
 1. [BucketIterator](#)
 7. [BatchWrapper](#)
 8. [Training a text classifier](#)
 1. [Initializing the model](#)
 2. [Putting the pieces together again](#)
 9. [Training loop](#)
 10. [Prediction mode](#)
 1. [Converting predictions into a pandas DataFrame](#)
 5. [Summary](#)
 12. [Building your Own Chatbot](#)
 1. [Why chatbots as a learning example?](#)
 1. [Why build a chatbot?](#)
 2. [Quick code means word vectors and heuristics](#)
 1. [Figuring out the right user intent](#)
 1. [Use case – food order bot](#)
 2. [Classifying user intent](#)
 3. [Bot responses](#)
 1. [Better response personalization](#)
 3. [Summary](#)
 13. [Web Deployments](#)
 1. [Web deployments](#)
 1. [Model persistence](#)
 2. [Model loading and prediction](#)
 3. [Flask for web deployments](#)
 2. [Summary](#)