

Continue



Turning text into valuable information is essential for many businesses looking to gain a competitive advantage. There have been many improvements in natural language processing and users have a lot of options when choosing to work on a problem. However, it's not always clear which NLP tools or libraries would work for a business use—or which techniques you should use and in what order. This practical book provides theoretical background and real-world case studies with detailed code examples to help developers and data scientists obtain insight from text online. Authors Jens Albrecht, Sidharth Ramachandran, and Christian Winkler use blueprints for text-related problems that apply state-of-the-art machine learning methods in Python. If you have a fundamental understanding of statistics and machine learning along with basic programming experience in Python, you're ready to get started. You'll learn how to: Crawl and clean then explore and visualize textual data in different formats Preprocess and vectorize text for machine learning Apply methods for classification, topic analysis, summarization, and knowledge extraction Use semantic word embeddings and deep learning approaches for complex problems Work with Python NLP libraries like spaCy, NLTK, and Gensim in combination with scikit-learn, Pandas, and PyTorch Jens Albrecht, Sidharth Ramachandran, Christian Winkler Published by O'Reilly, 2020 Find the book at O'Reilly Amazon.com Amazon.de Amazon.co.uk Amazon.fr Amazon.in If you like the book or the code examples here, please leave a friendly comment on Amazon! Free download of Chapter 7 "How to Explain a Classifier". This repository is currently in preparation. Please do not yet send any comments. This repository contains the code examples of our O'Reilly book. You will find a subdirectory for each chapter containing a Jupyter notebook and additional files for the setup. Below you find the links to view the notebooks here on GitHub or execute them directly on Google Colab. In the section thereafter you will find instructions to setup the environment on your local computer. If you discover any problems or have recommendations on how to improve the code, do not hesitate to create an issue here in the repository. For errors in the book text, please use O'Reilly's errata page. spaCy 3.0 and Gensim 4.0 The book uses spaCy 2.3.2 and gensim 3.8.3. spaCy 3.0 is now officially release with several new features and a few API changes (. Gensim 4.0 is in beta (. We are already updating our notebooks. But currently textacy is not yet supporting spaCy 3.0, although work is already in progress (see this pull request from us). Until textacy for spaCy 3.0 is released, you can use our own fork for the installation (see blueprints.yaml in this directory). For each chapter of the book we provide three links: "git" opens the notebook for viewing here on GitHub (sometimes not working because of GitHub issue) "nbviewer" opens the notebook for viewing on nbviewer.ipynb.org "colab" opens a runnable copy on Google's Colab service If you run the notebook locally or on Colab, you can execute each cell separately by hitting "Shift-enter". Do not skip cells and don't forget to run the first code cells for the setup. The following instructions should work on Linux, Windows and MacOS. If you are a Windows user familiar with Linux, you should check out the Windows Subsystem for Linux, Version 2 (WSL2). This allows to use a Linux system on the Windows machine. However, using native Windows should also be no problem. It is helpful to install git on your machine, but you can also download the full repository from GitHub as a zip file. If you use git, run the following commands from the command line: git clone cd blueprints-text Otherwise download the zip file, unpack it to a location convenient to you, and open a command line terminal in the project directory blueprints-text. For local setup, we recommend to use Miniconda, a minimal version of the popular Anaconda distribution that contains only the package manager conda and Python. Follow the installation instructions on the Miniconda Homepage. If you already have Anaconda or Miniconda installed on your system: That's fine. We will create a separate virtual environment for the blueprints book so that our installation will not interfere with your previous setup. After installation of Anaconda/Miniconda, run the following command(s) from the project directory: conda env create --name blueprints --file blueprints.yml conda activate blueprints The prompt should change after activation and indicate that you are working in the blueprints environment. Our installation includes the Jupyter notebook extensions. We suggest to enable the extensions "table of contents" (toc2), "execute time", and "variable inspector" (varInspector): jupyter nbextension enable toc2/main jupyter nbextension enable execute_time/ExecuteTime jupyter nbextension enable varInspector/main Now you can start the Jupyter notebook server: If working on WSL under Windows, add --no-browser. Browse to the respective chapter and open the notebook file (suffix .ipynb) Jens Albrecht is a full-time professor for Computer Science Department at the Nuremberg Institute of Technology. His work focuses on data management and analytics with a focus on text. He holds a doctorates degree in computer science. Before he rejoined academia in 2012, he has been working for over a decade in the industry as consultant and data architect. He is author of several articles on Big Data management and analysis. Sidharth Ramachandran currently leads a team of data scientists at GIG helping to build data products for the consumer goods industry. He has over 10 years of experience in software engineering and data science across telecom, banking and marketing industries. Sidharth also co-founded WACAO, a smart personal assistant on Whatsapp which was also featured on Techcrunch. He holds an undergraduate engineering degree from IIT Roorkee and an MBA from IIM Kozhikode. Sidharth is passionate about solving real problems through technology and loves to hack through personal projects in his free time. Christian Winkler is a Data Scientist and Machine Learning Architect. He holds a PhD in theoretical physics and has been working in the field of large data volumes and artificial intelligence for 20 years, with particular focus on scalable systems and intelligent algorithms for mass text processing. He is founder of datanizing GmbH, speaker at conferences and author of Machine Learning / Text Analytics articles. Image not available forColour: To view this video download Flash Player Jens Albrecht, Sidharth Ramachandran, Christian Winkler Published by O'Reilly, 2020 Find the book at O'Reilly Amazon.com Amazon.de Amazon.co.uk Amazon.fr Amazon.in If you like the book or the code examples here, please leave a friendly comment on Amazon! Free download of Chapter 7 "How to Explain a Classifier". This repository is currently in preparation. Please do not yet send any comments. This repository contains the code examples of our O'Reilly book. You will find a subdirectory for each chapter containing a Jupyter notebook and additional files for the setup. Below you find the links to view the notebooks here on GitHub or execute them directly on Google Colab. In the section thereafter you will find instructions to setup the environment on your local computer. If you discover any problems or have recommendations on how to improve the code, do not hesitate to create an issue here in the repository. For errors in the book text, please use O'Reilly's errata page. spaCy 3.0 and Gensim 4.0 The book uses spaCy 2.3.2 and gensim 3.8.3. spaCy 3.0 is now officially release with several new features and a few API changes (. Gensim 4.0 is in beta (. We are already updating our notebooks. But currently textacy is not yet supporting spaCy 3.0, although work is already in progress (see this pull request from us). Until textacy for spaCy 3.0 is released, you can use our own fork for the installation (see blueprints.yaml in this directory). For each chapter of the book we provide three links: "git" opens the notebook for viewing here on GitHub (sometimes not working because of GitHub issue) "nbviewer" opens the notebook for viewing on nbviewer.ipynb.org "colab" opens a runnable copy on Google's Colab service If you run the notebook locally or on Colab, you can execute each cell separately by hitting "Shift-enter". Do not skip cells and don't forget to run the first code cells for the setup. The following instructions should work on Linux, Windows and MacOS. If you are a Windows user familiar with Linux, you should check out the Windows Subsystem for Linux, Version 2 (WSL2). This allows to use a Linux system on the Windows machine. However, using native Windows should also be no problem. It is helpful to install git on your machine, but you can also download the full repository from GitHub as a zip file. If you use git, run the following commands from the command line: git clone cd blueprints-text Otherwise download the zip file, unpack it to a location convenient to you, and open a command line terminal in the project directory blueprints-text. For local setup, we recommend to use Miniconda, a minimal version of the popular Anaconda distribution that contains only the package manager conda and Python. Follow the installation instructions on the Miniconda Homepage. If you already have Anaconda or Miniconda installed on your system: That's fine. We will create a separate virtual environment for the blueprints book so that our installation will not interfere with your previous setup. After installation of Anaconda/Miniconda, run the following command(s) from the project directory: conda env create --name blueprints --file blueprints.yml conda activate blueprints The prompt should change after activation and indicate that you are working in the blueprints environment. Our installation includes the Jupyter notebook extensions. We suggest to enable the extensions "table of contents" (toc2), "execute time", and "variable inspector" (varInspector): jupyter nbextension enable toc2/main jupyter nbextension enable execute_time/ExecuteTime jupyter nbextension enable varInspector/main Now you can start the Jupyter notebook server: If working on WSL under Windows, add --no-browser. Browse to the respective chapter and open the notebook file (suffix .ipynb) All download options have the same file, and should be safe to use. That said, always be cautious when downloading files from the internet, especially from sites external to Anna's Archive. For example, be sure to keep your devices updated. Help out the community by reporting the quality of this file: 0) A "file MD5" is a hash that gets computed from the file contents, and is reasonably unique based on that content. All shadow libraries that we have indexed on here primarily use MD5s to identify files. A file might appear in multiple shadow libraries. For information about the various datasets that we have compiled, see the Datasets page. For information about this particular file, check out its JSON file. Live/debug JSON version. Live/debug page. Turning text into valuable information is essential for businesses looking to gain a competitive advantage. With recent improvements in natural language processing (NLP), users now have many options for solving complex challenges. But it's not always clear which NLP tools or libraries would work for a business's needs, or which techniques you should use and in what order. This practical book provides data scientists and developers with blueprints for best practice solutions to common tasks in text analytics and natural language processing. Authors Jens Albrecht, Sidharth Ramachandran, and Christian Winkler provide real-world case studies and detailed code examples in Python to help you get started quickly. Extract data from APIs and web pages Prepare textual data for statistical analysis and machine learning Use machine learning for classification, topic modeling, and summarization Explain AI models and classification results Explore and visualize semantic similarities with word embeddings Identify customer sentiment in product reviews Create a knowledge graph based on named entities and their relations It looks like you're offline. December 24, 2023 Edited by Quarticle //covers.openlibrary.org/b/id/14559594-S.jpg December 20, 2023 Edited by ImportBot import existing book September 30, 2021 Edited by ImportBot import existing book August 26, 2020 Created by ImportBot Imported from Better World Books record You can't perform that action at this time. Get full access to Blueprints for Text Analytics Using Python and 60K+ other titles, with a free 10-day trial of O'Reilly. There are also live events, courses curated by job role, and more. Turning text into valuable information is essential for businesses looking to gain a competitive advantage. With recent improvements in natural language processing (NLP), users now have many options for solving complex challenges. But it's not always clear which NLP tools or libraries would work for a business's needs, or which techniques you should use and in what order. This practical book provides data scientists and developers with blueprints for best practice solutions to common tasks in text analytics and natural language processing. Authors Jens Albrecht, Sidharth Ramachandran, and Christian Winkler provide real-world case studies and detailed code examples in Python to help you get started quickly. Extract data from APIs and web pages Prepare textual data for statistical analysis and machine learning Use machine learning for classification, topic modeling, and summarization Explain AI models and classification results Explore and visualize semantic similarities with word embeddings Identify customer sentiment in product reviews Create a knowledge graph based on named entities and their relations Jens Albrecht, Sidharth Ramachandran, Christian Winkler Published by O'Reilly, 2020 Find the book at O'Reilly Amazon.com Amazon.de Amazon.co.uk Amazon.fr Amazon.in If you like the book or the code examples here, please leave a friendly comment on Amazon! Free download of Chapter 7 "How to Explain a Classifier". This repository is currently in preparation. Please do not yet send any comments. This repository contains the code examples of our O'Reilly book. You will find a subdirectory for each chapter containing a Jupyter notebook and additional files for the setup. Below you find the links to view the notebooks here on GitHub or execute them directly on Google Colab. In the section thereafter you will find instructions to setup the environment on your local computer. If you discover any problems or have recommendations on how to improve the code, do not hesitate to create an issue here in the repository. For errors in the book text, please use O'Reilly's errata page. spaCy 3.0 and Gensim 4.0 The book uses spaCy 2.3.2 and gensim 3.8.3. spaCy 3.0 is now officially release with several new features and a few API changes (. Gensim 4.0 is in beta (. We are already updating our notebooks. But currently textacy is not yet supporting spaCy 3.0, although work is already in progress (see this pull request from us). Until textacy for spaCy 3.0 is released, you can use our own fork for the installation (see blueprints.yaml in this directory). For each chapter of the book we provide three links: "git" opens the notebook for viewing here on GitHub (sometimes not working because of GitHub issue) "nbviewer" opens the notebook for viewing on nbviewer.ipynb.org "colab" opens a runnable copy on Google's Colab service If you run the notebook locally or on Colab, you can execute each cell separately by hitting "Shift-enter". Do not skip cells and don't forget to run the first code cells for the setup. The following instructions should work on Linux, Windows and MacOS. If you are a Windows user familiar with Linux, you should check out the Windows Subsystem for Linux, Version 2 (WSL2). This allows to use a Linux system on the Windows machine. However, using native Windows should also be no problem. It is helpful to install git on your machine, but you can also download the full repository from GitHub as a zip file. If you use git, run the following commands from the command line: git clone cd blueprints-text Otherwise download the zip file, unpack it to a location convenient to you, and open a command line terminal in the project directory blueprints-text. For local setup, we recommend to use Miniconda, a minimal version of the popular Anaconda distribution that contains only the package manager conda and Python. Follow the installation instructions on the Miniconda Homepage. If you already have Anaconda or Miniconda installed on your system: That's fine. We will create a separate virtual environment for the blueprints book so that our installation will not interfere with your previous setup. After installation of Anaconda/Miniconda, run the following command(s) from the project directory: conda env create --name blueprints --file blueprints.yml conda activate blueprints The prompt should change after activation and indicate that you are working in the blueprints environment. Our installation includes the Jupyter notebook extensions. We suggest to enable the extensions "table of contents" (toc2), "execute time", and "variable inspector" (varInspector): jupyter nbextension enable toc2/main jupyter nbextension enable execute_time/ExecuteTime jupyter nbextension enable varInspector/main Now you can start the Jupyter notebook server: If working on WSL under Windows, add --no-browser. Browse to the respective chapter and open the notebook file (suffix .ipynb)