VA | U.S. Department of Veterans Affairs

Using Python in VINCI

VA Informatics and Computing Infrastructure

Overview

- Basic Python Introduction
  - What is Python?
  - Why use Python?
- How to use Python in VINCI
  - Requirements
  - Installing Python
  - Adding packages
- Python use cases in VINCI
  - Projects using Python
  - Python packages developed in VINCI

Overview

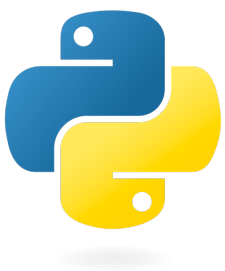
- **Basic Python Introduction**
  - **What is Python?**
  - **Why use Python?**

- How to use Python in VINCI
  - Requirements
  - Installing Python
  - Adding packages

- Python use cases in VINCI
  - Projects using Python
  - Python packages developed in VINCI

What is Python?

- Multi-paradigm programming language
  - Object Oriented Programming
  - Functional Programming
  - etc.
- Simple syntax
  - Control flow determined by whitespace
  - Dynamic typing
- Open source, community funded development
- First released in 1991



python™

```
def gcd(a, b):
    if(b == 0):
        return a
    else:
        return gcd(b, a % b)

print(gcd(60, 48))

>>> 12
```

Why use Python? – The Language

- Emphasizes code readability
  - Improves research transparency
  - Good for reproducibility
- Platform agnostic
  - The same code works anywhere
- Allows for quickly going from idea to working prototype

Why use Python? – The Community

- Python is a primary language for data science community
- Thousands of available packages for different algorithms or tools are maintained
  - Utilize work of others rather than creating implementations from scratch
- Community help-sites like StackOverflow and Github Issues provide quick troubleshooting

Overview

- Basic Python Introduction
  - What is Python?
  - Why use Python?

- **How to use Python in VINCI**
  - **Requirements**
  - **Installing Python**
  - **Adding packages**

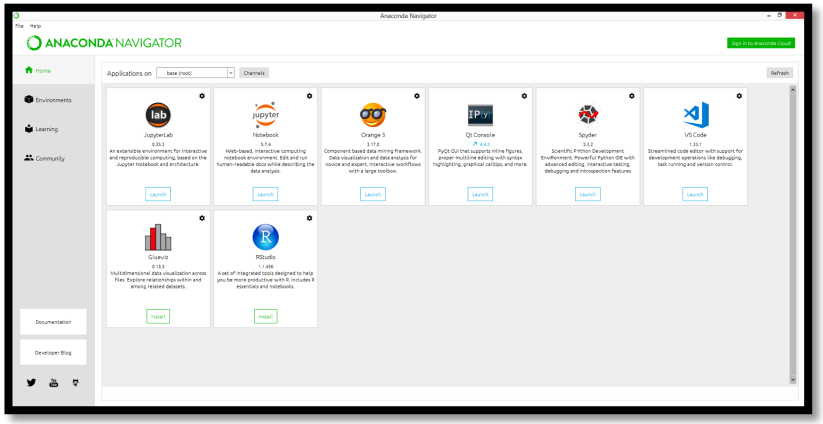
- Python use cases in VINCI
  - Projects using Python
  - Python packages developed in VINCI

Requirements for using Python

- Development Workspace access (not Standard Workspace)
  - R is a good substitute for Python if only the Standard Workspace is available
  - Apply to get a Development VM at: VINCI Central -> Computing Cloud
- Access to the S: Drive with approved software
- Install Anaconda from installers available on S: drive

What is Anaconda?

- Python (and R) distribution
- Designed for scientific computing and large-scale applications
- Manages virtual environments for projects
- Handles package installation
- Provides a user interface and command line tools

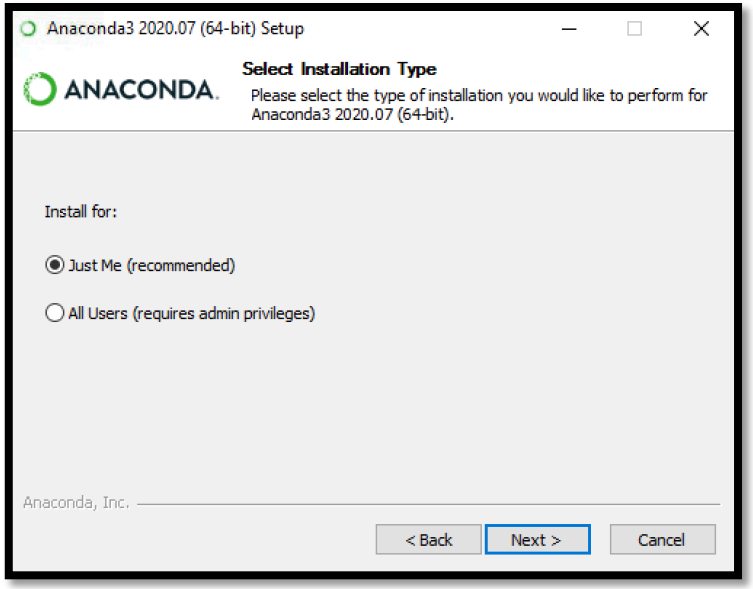


Why use Anaconda?

- Virtual environments protect your code
  - Keep packages separate
  - Allow different projects to use different versions of a package
  - Protect python from corruption
- Anaconda's Python distribution is faster than other distributions
- User interface allows users without command line experience to use Python

Installing Anaconda

- Anaconda installers available at S:\Site Licensed\Anaconda
- Anaconda can be installed ***without*** administrator privileges for one user at a time
- Detailed instructions available online and during installation



Packages Included in Anaconda

Adding New Packages to VINCI

- Anaconda includes many popular general-purpose packages
- Extra installation required for specialized packages
- Installing large packages is difficult
- Requires downloading installation files or project source code outside VINCI then uploading and installing within VINCI
- Requires downloading ***ALL*** requirements/dependencies of the package

Adding New Packages to VINCI – Options

Manual Download

- Manually download all packages and requirements
- Slow
- Prone to user error
- No command line necessary

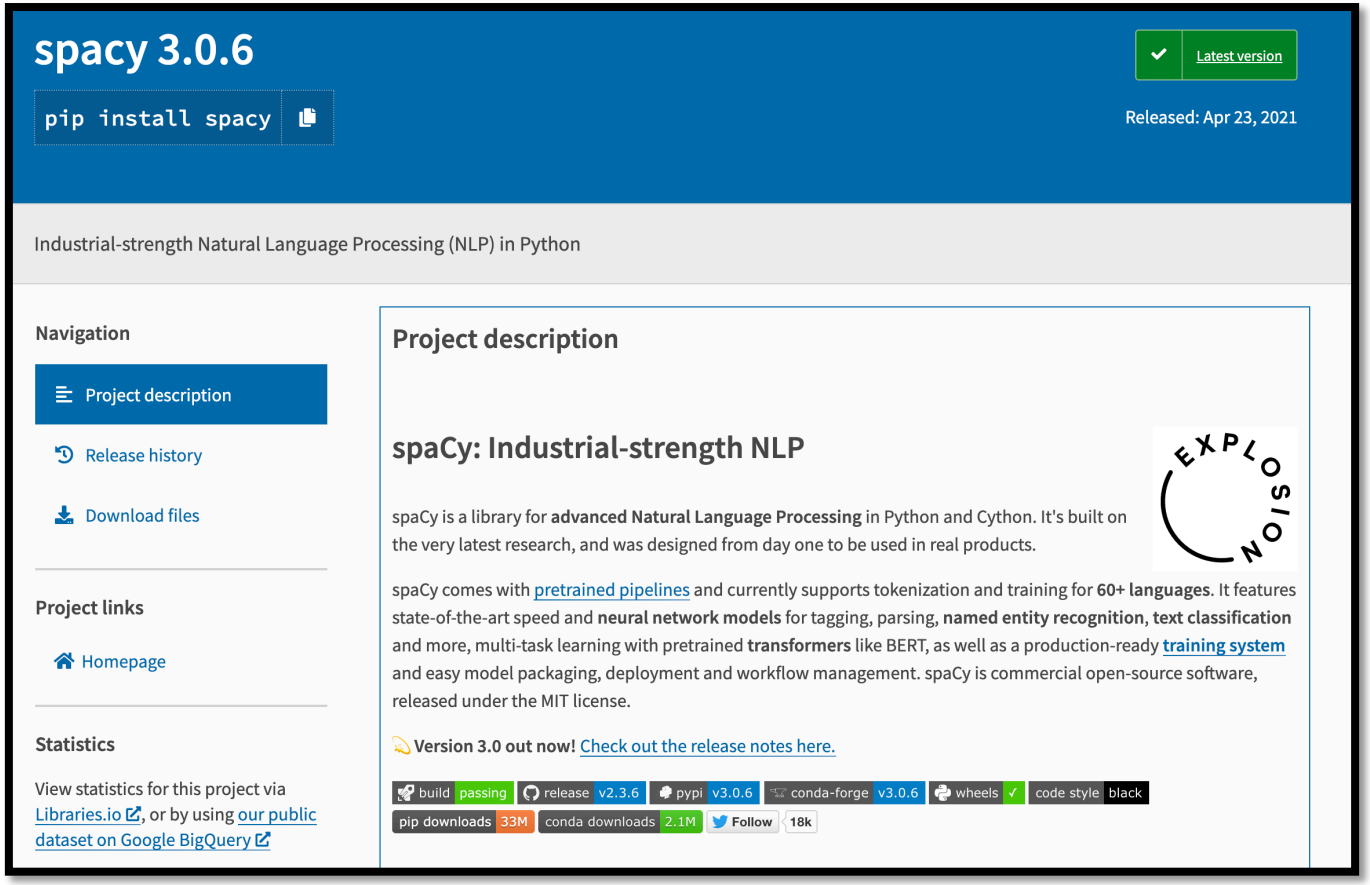
Pip Download

- Automatically download all packages and requirements
- Fast
- Sometimes downloads incompatible versions
- Using command line

Adding New Packages to VINCI – Manual Download

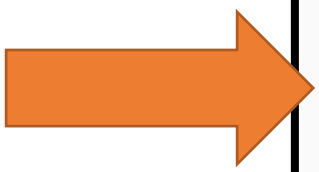
spaCy is a package you might want to install in VINCI.

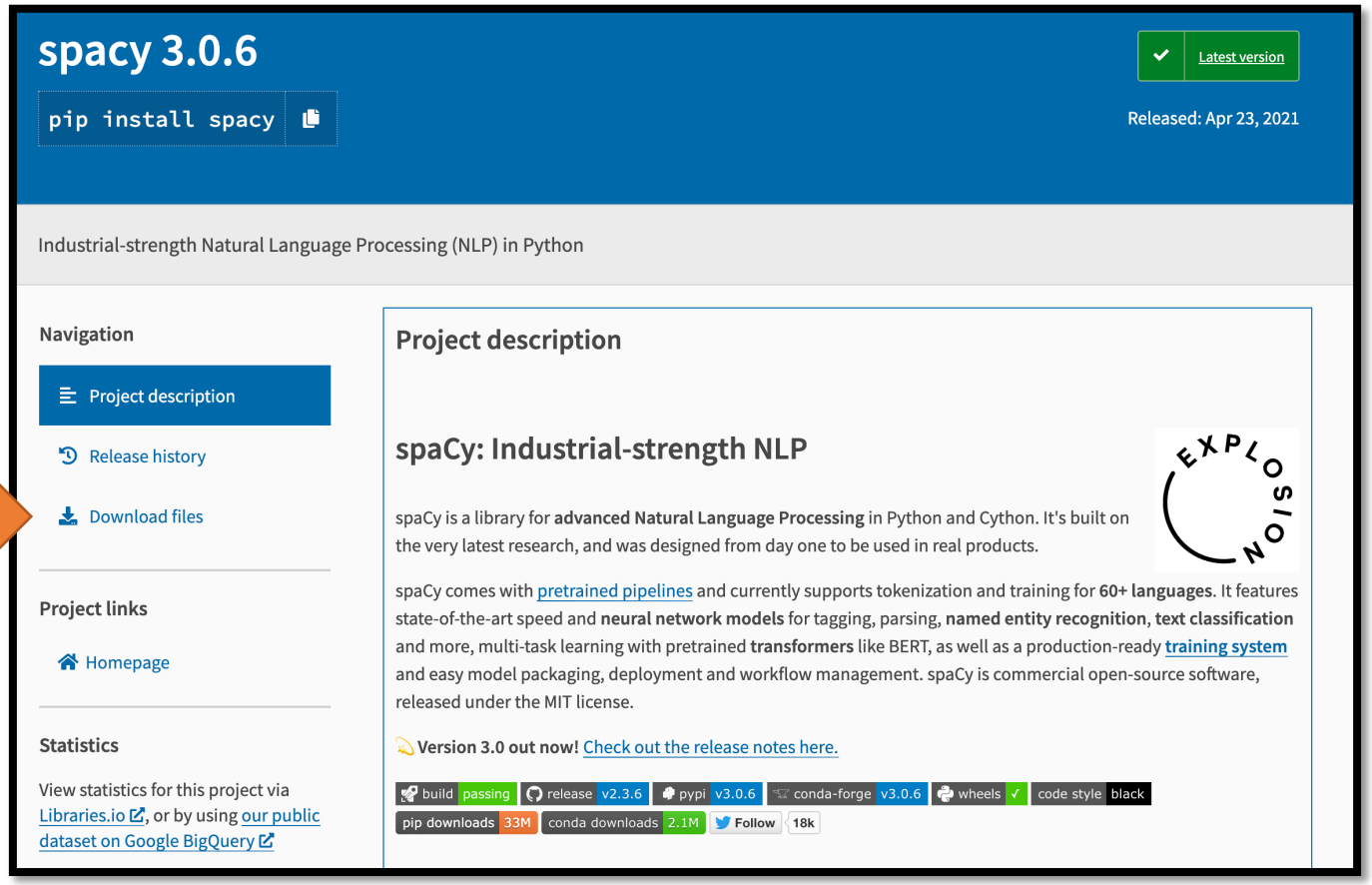
This is the PyPI webpage for spaCy.



Adding New Packages to VINCI – Manual Download

Click
Download Files





Adding New Packages to VINCI – Manual Download

Look for the files for your Python version.

cp36 = python 3.6
cp37 = python 3.7
cp38 = python 3.8
…

Adding New Packages to VINCI – Manual Download

Look at filenames to find a file compatible with your development workspace.

Any computer:
- `noarch`
- no platform specification

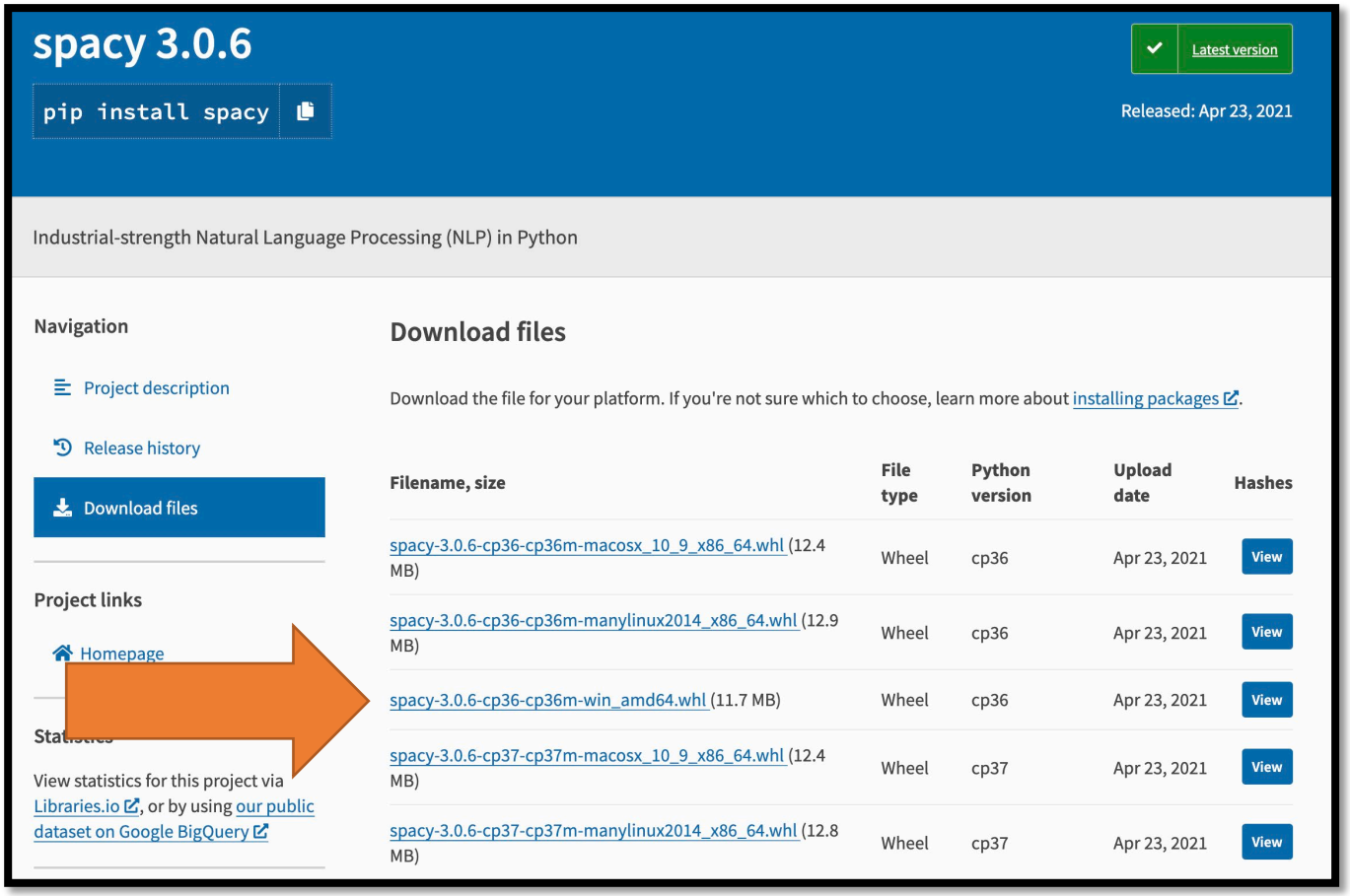
Windows:
- `win_amd64`

Linux:
- `manylinux`

Adding New Packages to VINCI – Manual Download

Download the file for your computer ***AND*** python version.

This file is for a Windows workspace using Python 3.6

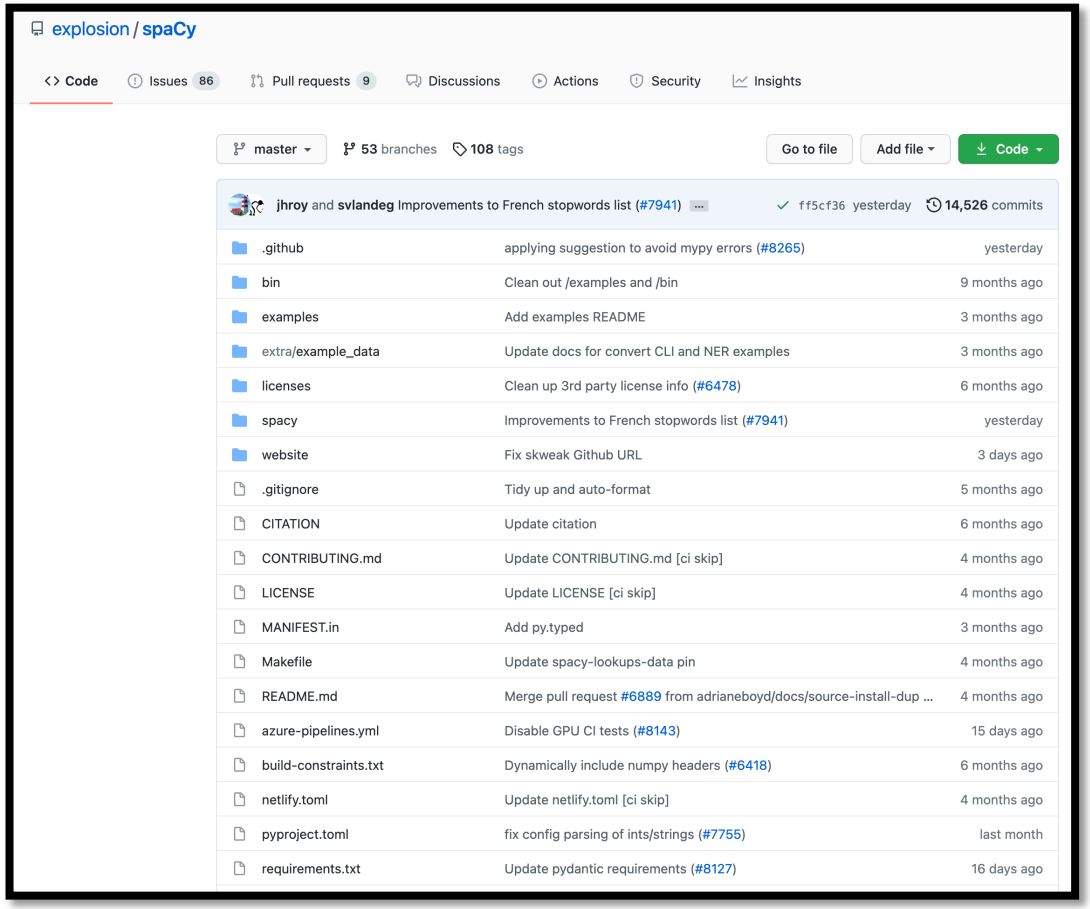


Adding New Packages to VINCI – Manual Download

Now to get dependencies.

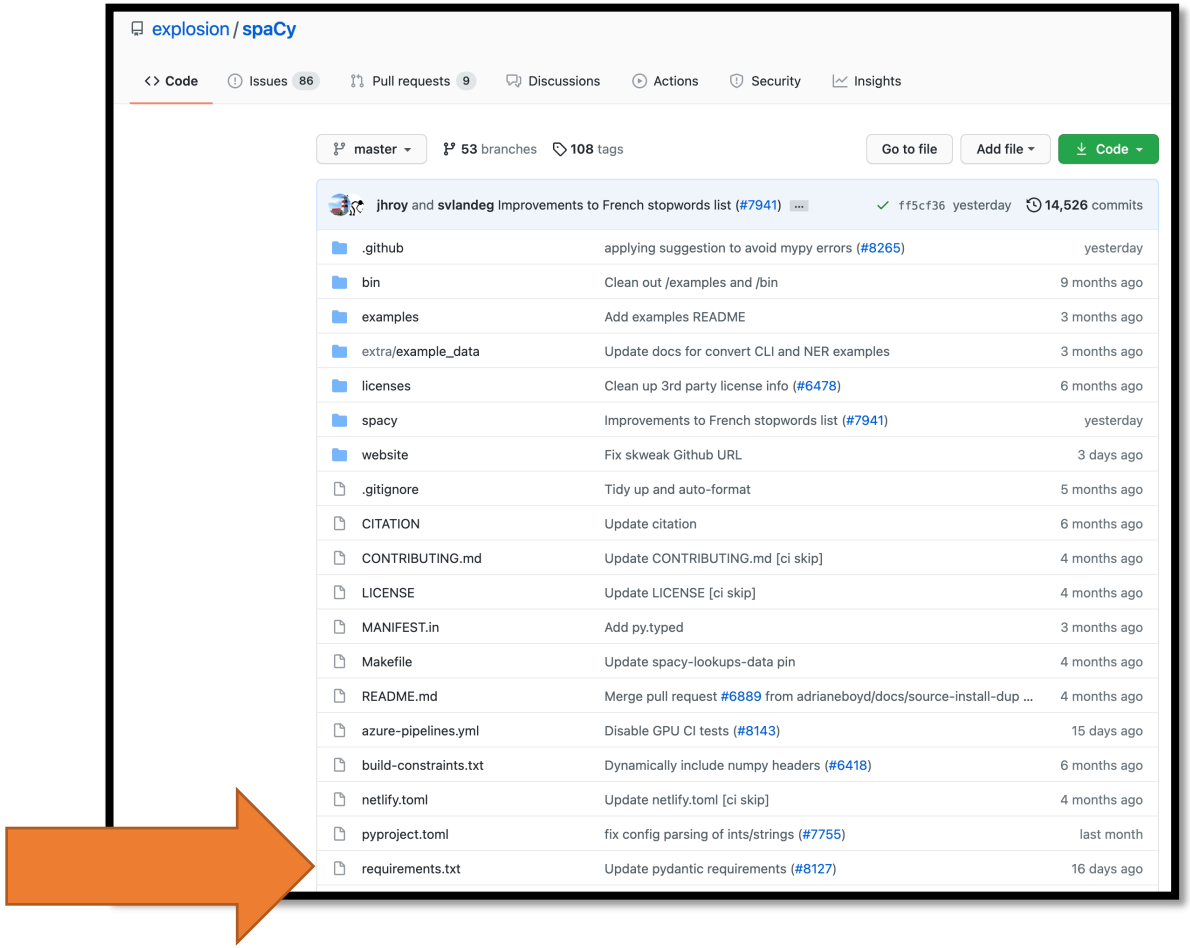
The easiest way to look at this is to open the Github page for the same project.

You can also open the file you just downloaded (but save a copy for later).



Adding New Packages to VINCI – Manual Download

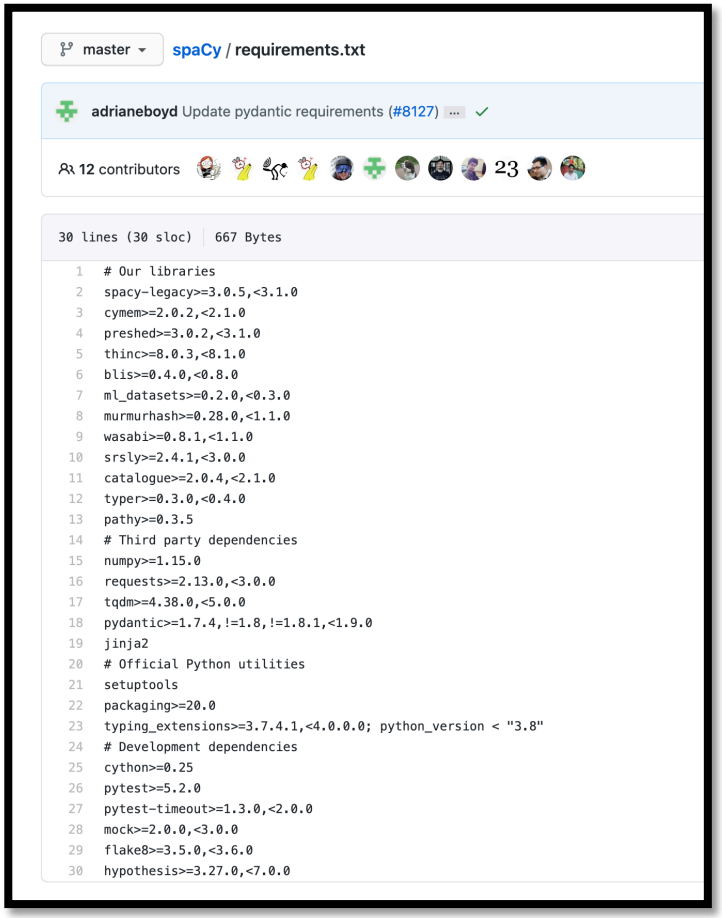
Open requirements.txt



Adding New Packages to VINCI – Manual Download

Download all listed packages ***AND*** their requirements using the same steps.

This process can be long for packages like spaCy with lots of features, developers, and users.



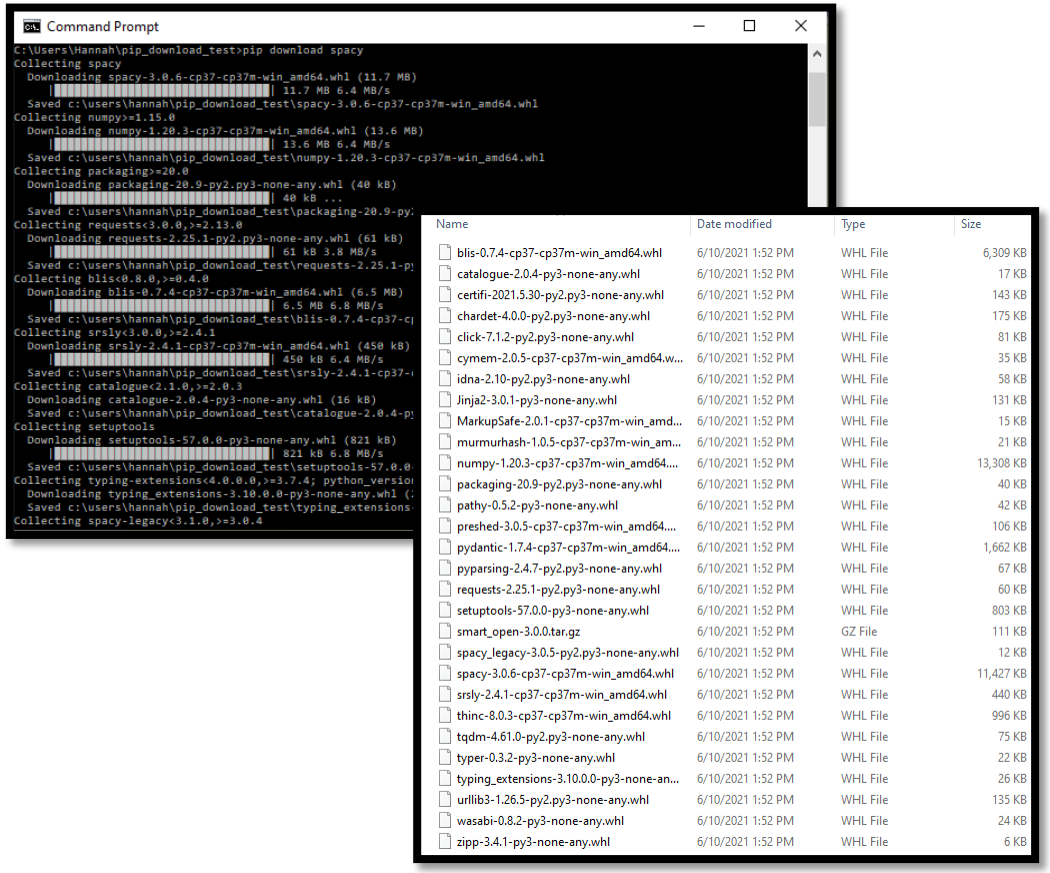
Adding New Packages to VINCI – Pip Download

Must use command line, no UI available for pip

```
pip download <your_package>
```

```
pip download spacy
```

Downloads the package and all requirements into the current location.

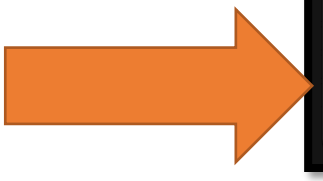


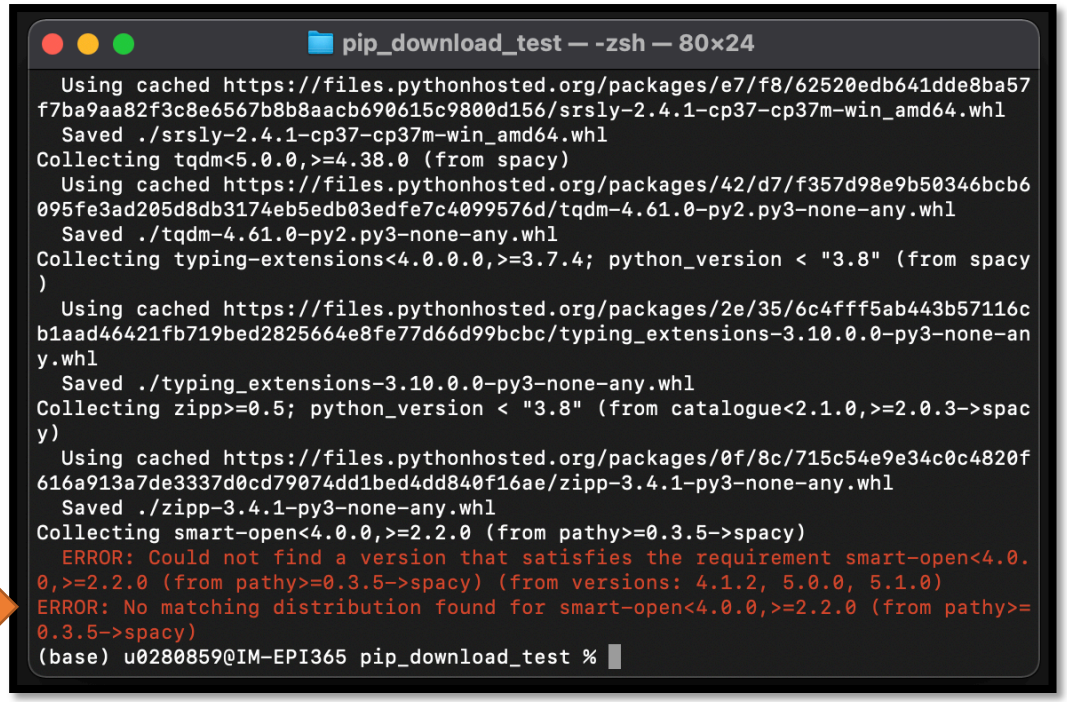
Adding New Packages to VINCI – Pip Download

Convenient ***ONLY*** when you have matching platforms.

Will have errors or download incorrect versions when downloading on a different operating system.

This is an error that appears when downloading windows spaCy on a Mac



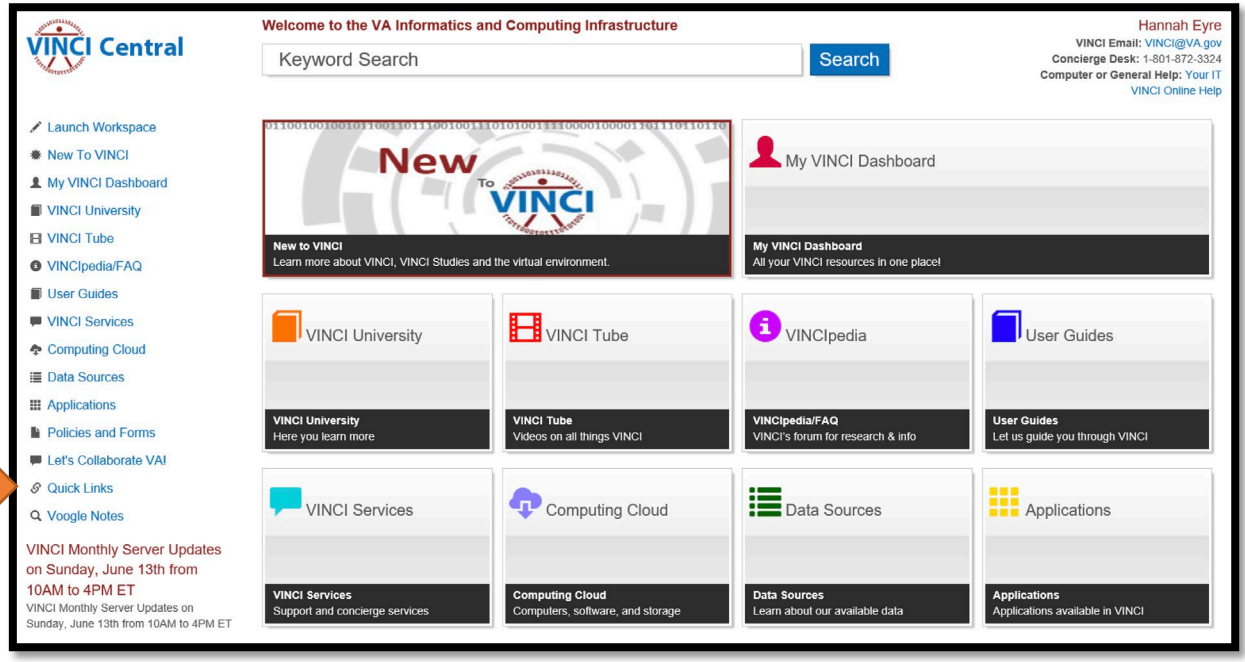


Adding New Packages to VINCI

Once all packages are downloaded

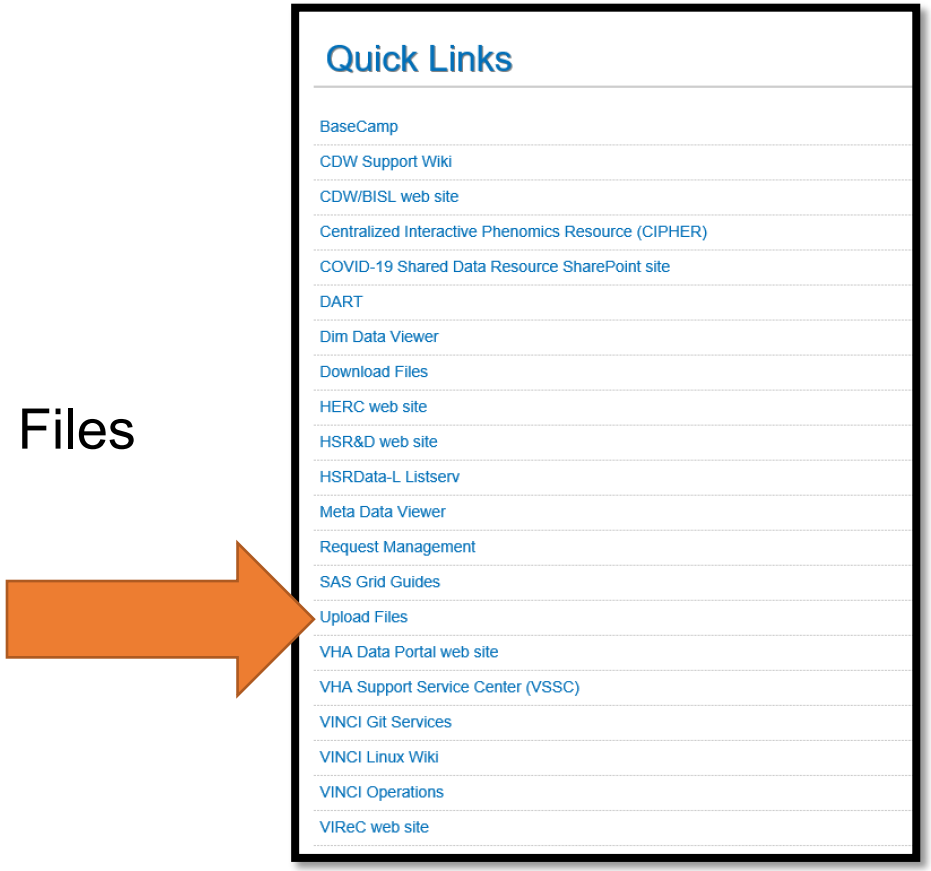
Navigate to VINCI Central

Click Quick Links



Adding New Packages to VINCI

Then click Upload Files

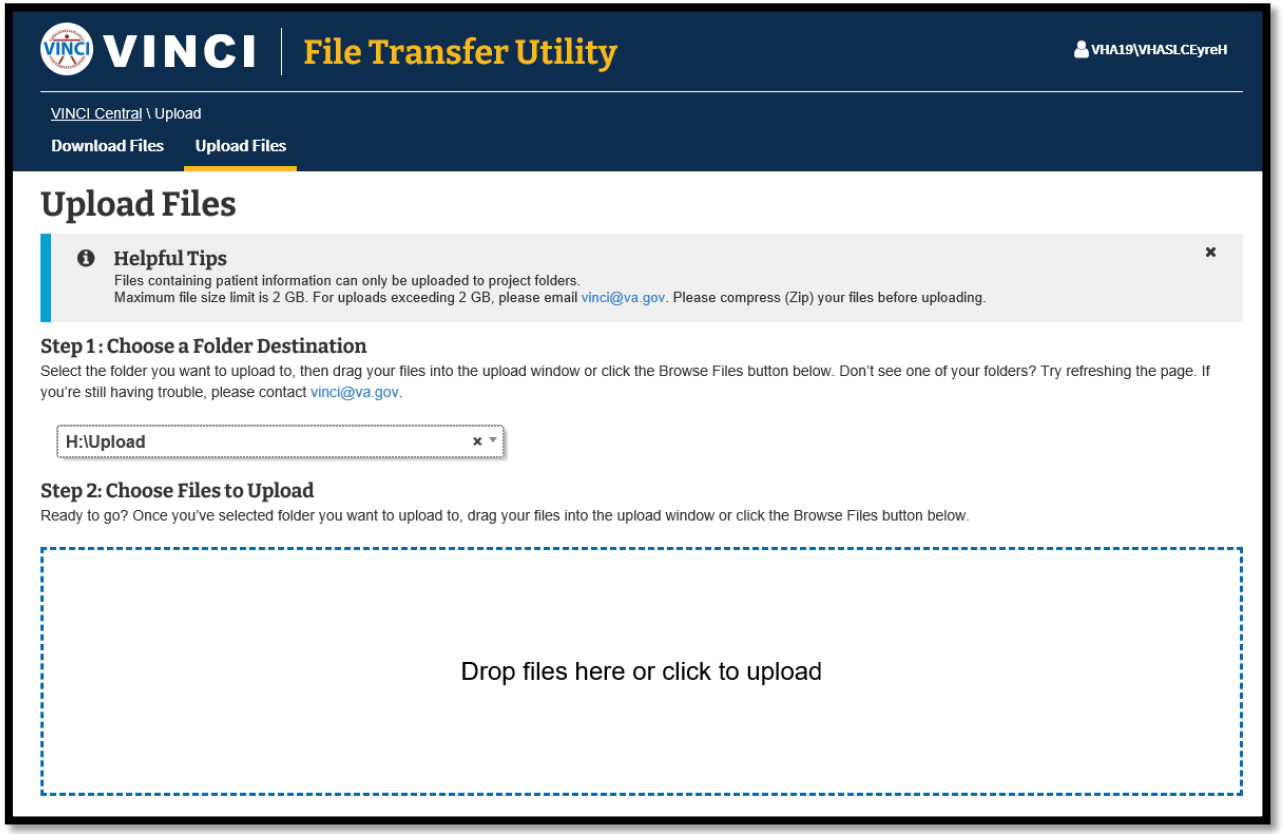


Adding New Packages to VINCI

Upload all downloaded packages with VINCI Upload tool.

Can be uploaded one at a time or zipped together

Pip install after uploading

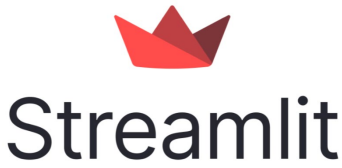


COMING SOON – VINCI Anaconda Environment

- A pre-made environment will be available on the S drive made by VINCI NLP team
- Usable on any Windows development workspace after installing Anaconda
- Unzip environment in your development workspace and use as-is
- Includes many packages beyond basic Anaconda packages
- Updates and support will be limited, but it can help skip upload process

COMING SOON – VINCI Anaconda Environment

Some of the additional packages included in the VINCI Anaconda Environment



How has Python already been used?

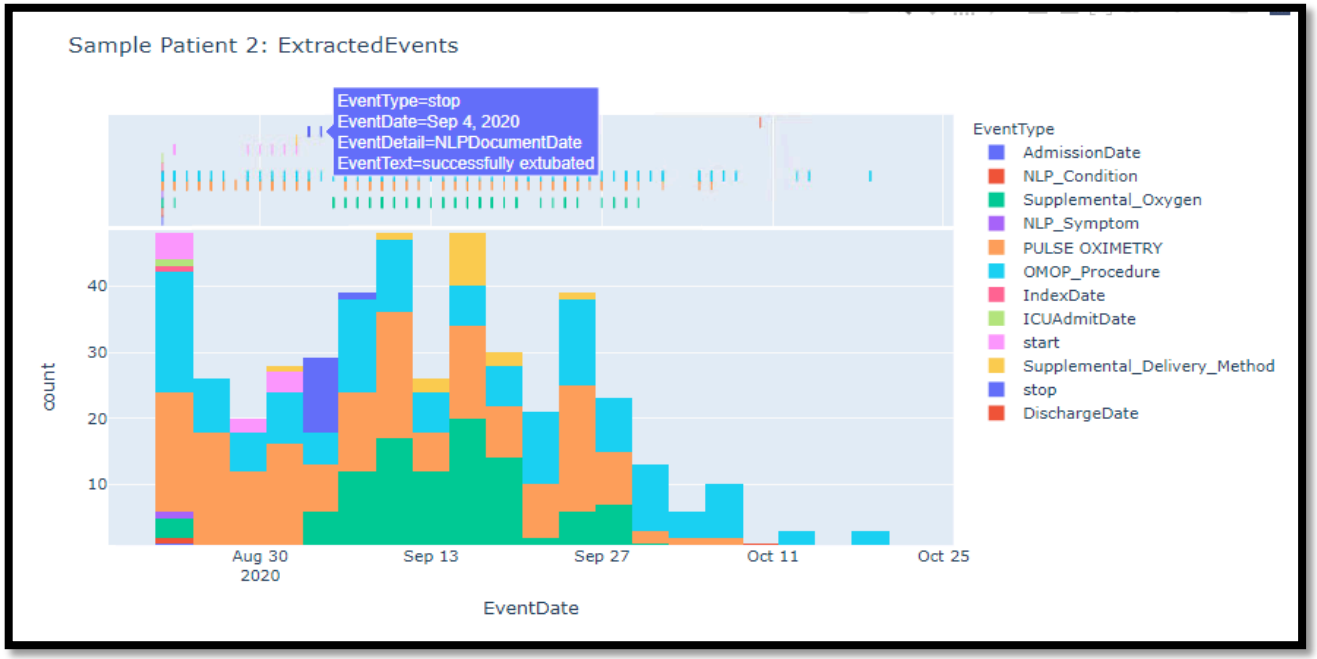
- Completed research and operational projects within VINCI
- Created datasets in the COVID-19 Shared Data Resource
- Communicated research results with interactive displays
- Developed open-source applications shared outside the VA

Data Visualization – Analysis

Visual representation can help debug or troubleshoot analysis code

Clear diagrams help understand the data intuitively

Can easily combine many data sources into one image

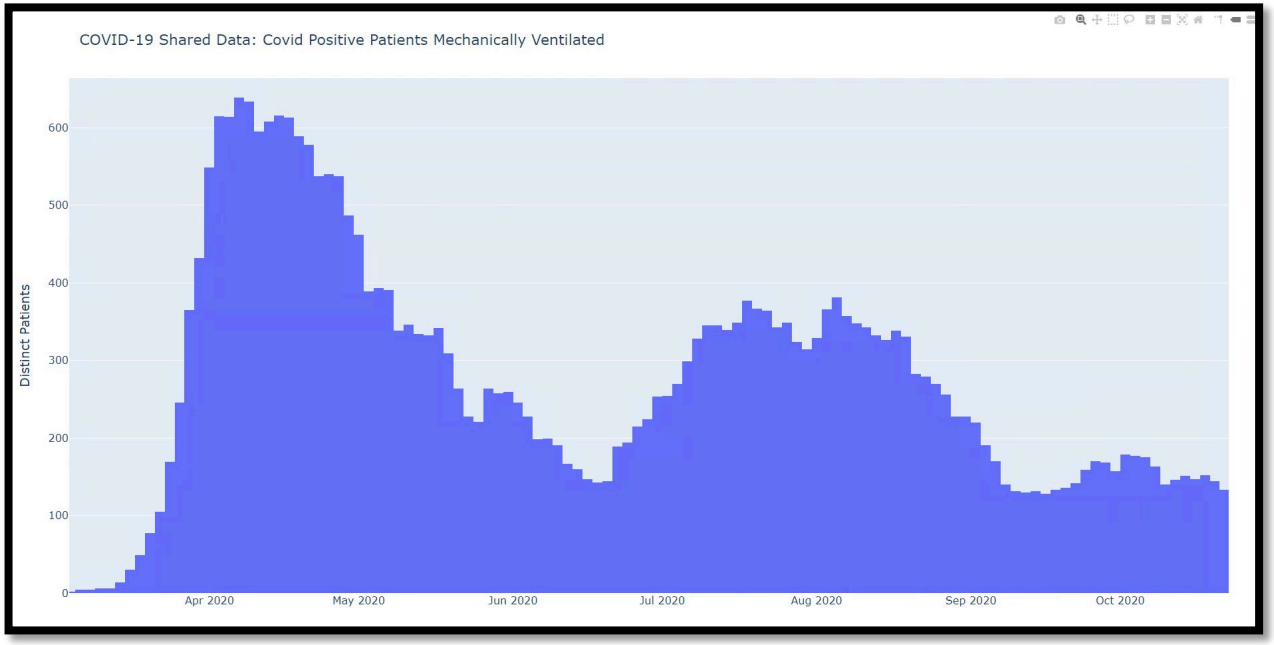


Data Visualization - Communication

Aggregated or cleaned up data used to communicate research results

Useful for dashboards that are understood by non-analyst team members

Results can be interactive



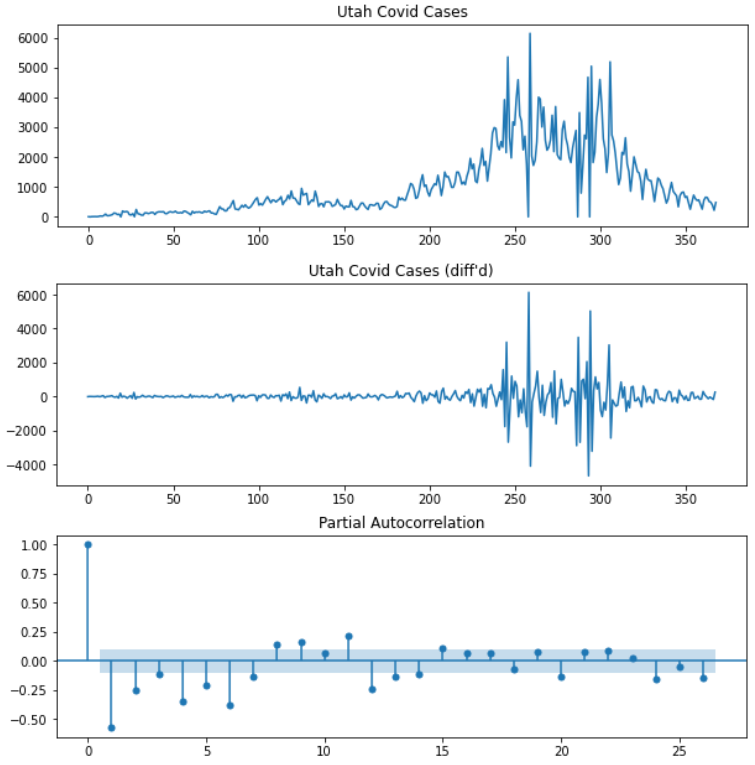
Data Science

Use mathematical and statistical techniques to understand VA data

Mathematical libraries allow powerful modeling techniques to be used

Optimized libraries allow for high performance on extremely large quantities of data

NumPy



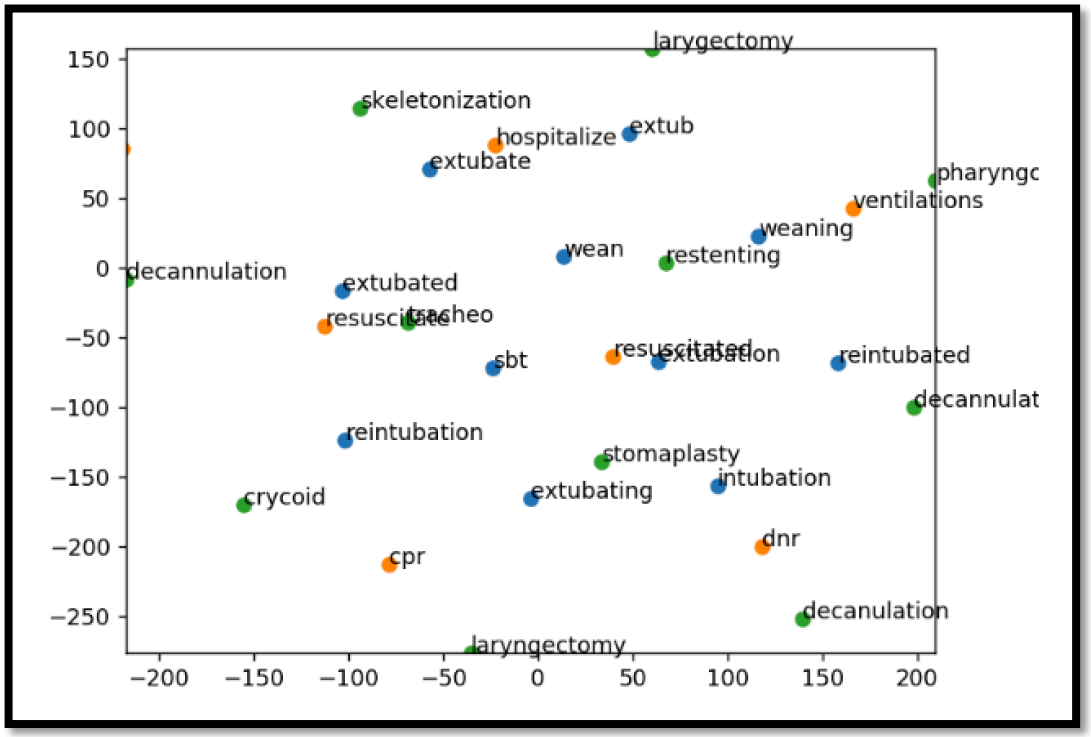
Natural Language Processing – Word Embeddings

Train word embeddings on raw VA clinical notes

Use up to hundreds of millions of notes

Use for text similarity and vocab expansion

Next word prediction in a sequence



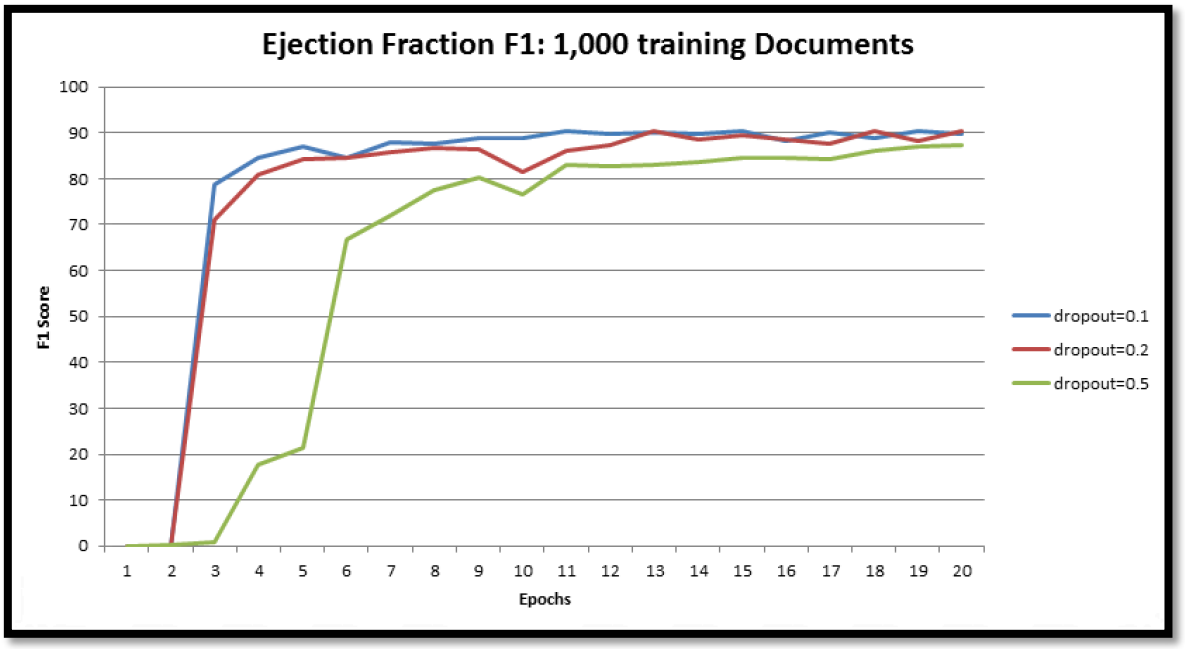
Natural Language Processing – Deep Learning

Train convolutional neural network models on a variety of tasks

Evaluate against a large test set

spaCy

Create deployment-ready models to apply on future data



Natural Language Processing - medspaCy

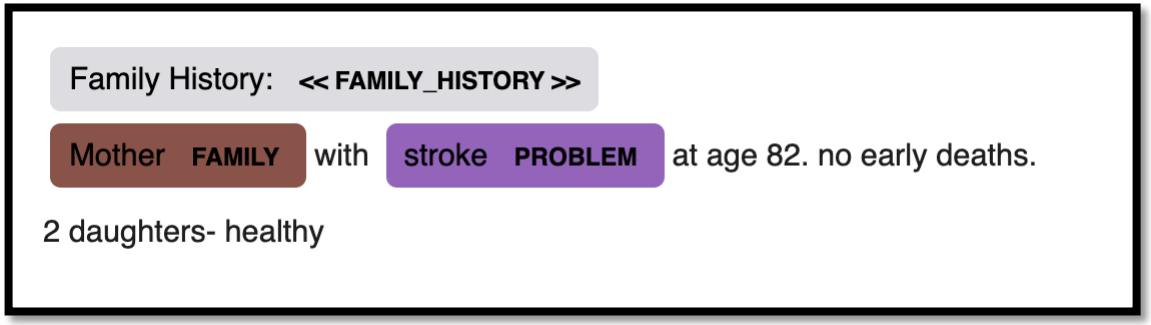
- medspaCy is an NLP toolkit based on spaCy
  - spaCy is the most popular general-purpose NLP package in Python
- Designed for performing common clinical NLP tasks in Python
- Compatible with other spaCy components
- Easy to incorporate other Python libraries for tasks like deep learning, visualization, and parallel deployment

Natural Language Processing - medspaCy

- medspaCy has several ways to identify clinical concepts
- Custom rule-based matching
- UMLS mapping with QuickUMLS
- Allows for use of spaCy's CNN models or models from any other source package

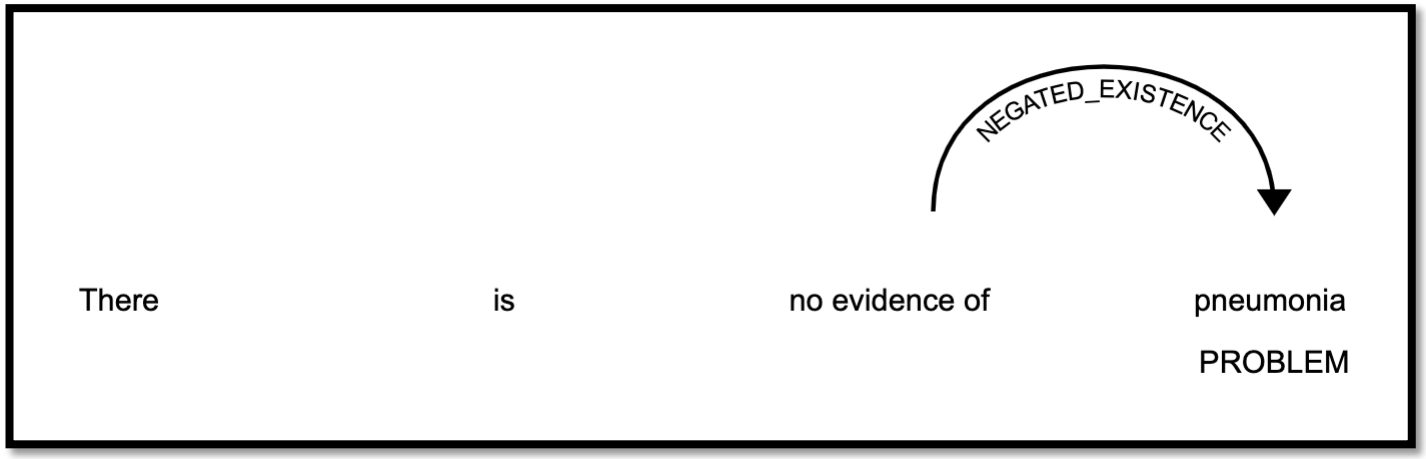
Natural Language Processing - medspaCy

- medspaCy offers an algorithm for clinical section detection
- Rule-based section patterns
- Large set of default rules available, but complete customization possible
- Can create section hierarchies for sections and subsections



Natural Language Processing - medspaCy

- medspaCy implements the ConText algorithm
- Allows for detection of common attributes such as negation, temporality, or experiencer
- Can be customized for special attributes as needed



Natural Language Processing - medspaCy

- medspaCy has many other utilities
- Pre- and Post-Processing rules
- Output to Pandas dataframes and databases
- Template pipelines for deploying with all reading/writing from SQL

Natural Language Processing - medspaCy

- medspacy is used to help create data in the COVID-19 Shared Data Resource
- Identification of COVID-19 positive VA patients for VA Biosurveillance
- Is incorporated into an R library called Clinspacy
- Used for classes at University of Melbourne
- Bio-NLP parser at University of Washington
- Available on GitHub and PyPI

Questions?

- Additional Trainings and Cyberseminars:
  - VINCI Central
  - Related HSR&D Cyberseminars
    - Introduction to NLP
    - Introduction to Python

- Other resources
  - Anaconda documentation online for using conda environments
  - StackOverflow/Github Issues for troubleshooting individual packages
  - Tutorials available through individual packages through Jupyter notebooks and youtube videos

- Helpdesk: VINCI@va.gov