

Instructions for invoiceReader project

1. create new folder
2. navigate to created folder
3. open cmd by typing in the Windows Explorer search field and navigate to this folder
4. create a virtual environment typing (python -m venv <name of venv like InvoiceReader>)
5. activate your virtual environment while you're in your root folder typing
(.\InvoiceReader\Scripts\activate)
6. install required packages creating requirements.txt and typing the required packages like this:

numpy

pandas

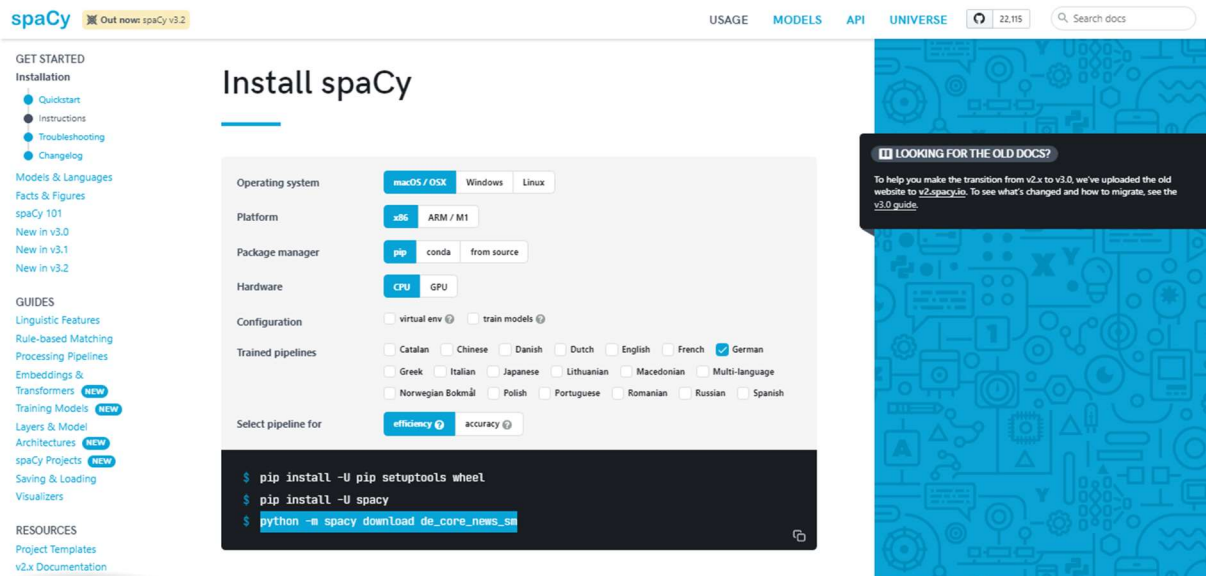
scipy

matplotlib

pillow

opencv-python

jupyter
7. now install the packages typing (pip install -r requirements.txt)
8. install tesseract-OCR, set environment variable and
9. see these pages and read for more info
 - (<https://tesseract-ocr.github.io/tessdoc/Data-Files.html>)
 - (<https://tesseract-ocr.github.io/tessdoc/Installation.html>)only in case, if you want to do OCR operations in other languages than English.
10. consider to download the tessdata, it's needed for language specific trained data in our case deu.traineddata from
 - (https://github.com/tesseract-ocr/tessdata_best)
11. now install pytesseract typing (pip install pytesseract)
12. visit (<https://spacy.io/usage>) and set the following configurations in order to download German specific spacy.



13. Pay attention to activating your virtual environment in your root directory or folder, means navigate to your root directory and type cmd instead of the path in the Windows explorer and after that activate your venv as described above.

14. now type

(`pip install -U spacy`) to install spacy and then

(`python -m spacy download de_core_news_sm`) → to install German pre-trained model

15. convert all pdf invoices to images using (PdfToImg_Converter1.py)

16. **Now create the 02_Data_Preparation.ipynb file in jupyter notebook and run the file as it is to prepare data for labeling and transforming into specific format. For more details see code comments!**

17. now transform manually your .csv file from step 16 to .txt file by saving the .csv file in (tab separated txt) format in Excel. Now put the saved .txt file to the root directory of your project where your .ipynb files are. **See data-74To95_text.txt** for more details on that in the root directory.

18. now run the 03_Data_Preprocessing.ipynb as the file name explains what to do. **See code comments for more info**

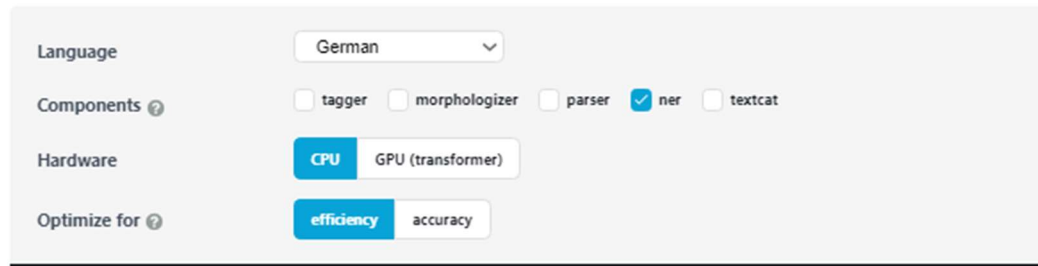
after processing your data you need to save your TrainData and TestData in pickle format, so **take at step 18**

19. now visit (<https://spacy.io/usage/training>) and then go to → Quickstart and set configs like this:

Quickstart NEW

The recommended way to train your spaCy pipelines is via the `spacy train` command on the command line. It only needs a single `config.cfg` configuration file that includes all settings and hyperparameters. You can optionally [overwrite](#) settings on the command line, and load in a Python file to register [custom functions](#) and architectures. This quickstart widget helps you generate a starter config with the **recommended settings** for your specific use case. It's also available in spaCy as the `init config` command.

Upgrade to the [latest version of spaCy](#) to use the quickstart widget. For earlier releases, follow the CLI instructions to generate a compatible config.

The image shows the spaCy Quickstart widget interface. It has a light gray background with several sections. The 'Language' section has a dropdown menu set to 'German'. The 'Components' section has checkboxes for 'tagger', 'morphologizer', 'parser', 'ner' (which is checked), and 'textcat'. The 'Hardware' section has two buttons: 'CPU' (highlighted in blue) and 'GPU (transformer)'. The 'Optimize for' section has two buttons: 'efficiency' (highlighted in blue) and 'accuracy'.

20. and then go to the right bottom corner and download this config file (base_config.cfg) and put to your target folder

21. now open the config file in jupyter notebook and at line 3 you see this command:

(`python -m spacy init fill-config ./base_config.cfg ./config.cfg`)

22. copy that command without parenthesis then click on → New and then → Terminal from jupyter notebook

23. now navigate to the folder in our case (1_InvoiceReaderProject) where the config file is and then

24. execute the command from step 21 and done !

```
PS C:\Users\Admin\Desktop\InvoiceReader_Project\1_InvoiceReaderProject> python -m spacy init fill-config ./base_config.cfg ./config.cfg
✓ Auto-filled config with all values
✓ Saved config
config.cfg
You can now add your data and train your pipeline:
python -m spacy train config.cfg --paths.train ./train.spacy --paths.dev ./dev.spacy
PS C:\Users\Admin\Desktop\InvoiceReader_Project\1_InvoiceReaderProject>
```

25. now visit this link again (<https://spacy.io/usage/training>) and scroll down to → Preparing Training Data and copy the code preprocess.py → make new file in jupyter note and rename to preprocess.py → paste the copied code from above there and edit the code as bellow:

```

1 import spacy
2 from spacy.tokens import DocBin
3 import pickle
4
5 nlp = spacy.blank("de")
6
7 # Load data
8 training_data = pickle.load(open('./data/TrainData.pickle','rb'))
9 testing_data = pickle.load(open('./data/TestData.pickle','rb'))
10
11 # the DocBin will store the example documents
12 db = DocBin()
13 for text, annotations in training_data:
14     doc = nlp(text)
15     ents = []
16     for start, end, label in annotations['entities']:
17         span = doc.char_span(start, end, label=label)
18         ents.append(span)
19     doc.ents = ents
20     db.add(doc)
21 db.to_disk("./data/train.spacy")
22
23 # the DocBin will store the example documents
24 db_test = DocBin()
25 for text, annotations in testing_data:
26     doc = nlp(text)
27     ents = []
28     for start, end, label in annotations['entities']:
29         span = doc.char_span(start, end, label=label)
30         ents.append(span)
31     doc.ents = ents
32     db_test.add(doc)
33 db_test.to_disk("./data/test.spacy")

```

26. create the folders output and data in your project folder, if you haven't yet

27. now go to your pre-opened Terminal in jupyter notebook and run → python .\preprocess.py

This file is from step 25 to remember And done, now we have saved our data in spacy format

28. if you haven't yet, create now the output directory and type the following command:

```
Python -m spacy train .\config.cfg --output .\output\ --paths.train .\data\train.spacy --paths.dev .\data\test.spacy
```

Make sure you see something like this output, this ist the training process for spacy NER-Model

```

===== Initializing pipeline =====
[2022-01-11 02:17:16,430] [INFO] Set up nlp object from config
[2022-01-11 02:17:16,457] [INFO] Pipeline: ['tok2vec', 'ner']
[2022-01-11 02:17:16,471] [INFO] Created vocabulary
[2022-01-11 02:17:16,475] [INFO] Finished initializing nlp object
[2022-01-11 02:17:16,962] [INFO] Initialized pipeline components: ['tok2vec', 'ner']
✓ Initialized pipeline

===== Training pipeline =====
i Pipeline: ['tok2vec', 'ner']
i Initial learn rate: 0.001
E      #      LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ---  -
  0         0          0.00    112.76    0.00    0.00    0.00    0.00
 66        200        465.81   2731.72    8.33   10.00    7.14    0.08
133        400         2.50     1.46    0.00    0.00    0.00    0.00
200        600        12.24     5.39    0.00    0.00    0.00    0.00
266        800         0.00     0.00    0.00    0.00    0.00    0.00
333       1000         0.00     0.00    0.00    0.00    0.00    0.00
400       1200         0.00     0.00    0.00    0.00    0.00    0.00
466       1400         0.15     0.06    0.00    0.00    0.00    0.00
533       1600        114.32    33.07    9.52   14.29    7.14    0.10
600       1800         11.85     2.32   23.53   66.67   14.29    0.24
666       2000        17.87     4.30   13.79   13.33   14.29    0.14
733       2200         0.00     0.00   13.79   13.33   14.29    0.14
815       2400         0.00     0.00   14.29   14.29   14.29    0.14
913       2600         0.00     0.00   14.29   14.29   14.29    0.14
1047      2800         0.00     0.00   14.29   14.29   14.29    0.14
1247      3000        203.81    19.20    9.52   14.29    7.14    0.10
1447      3200         4.55     0.86   10.00   16.67    7.14    0.10
1647      3400        30.36     4.55   14.81   15.38   14.29    0.15
✓ Saved pipeline to output directory
output\model-last
PS C:\Users\Admin\Desktop\InvoiceReader_Project\1_InvoiceReaderProject>

```

29. Now run the file 04_Predictions step by step in order to understand how the NER-Mode works.
See code comments for more.

This file is already cleaned and save as **predictions.py** for you. No need to save as .py file for predictions.

30. Last step → just run the file 05_Final_predictions.ipynb by opening and running step by step in jupyter notebook