```
import spacy
import pickle
import random


train_data = pickle.load(open('/content/train_data.pkl', 'rb'))
train_data[0]
```

```
('Govardhana K Senior Software Engineer  Bengaluru, Karnataka, Karnataka - Email me on Indeed: indeed.com/r/Govardhana-K/
b2de315d95905b68  Total IT experience 5 Years 6 Months Cloud Lending Solutions INC 4 Month • Salesforce Developer Oracle 5 Years
2 Month • Core Java Developer Languages Core Java, Go Lang Oracle PL-SQL programming, Sales Force Developer with APEX.
Designations & Promotions  Willing to relocate: Anywhere  WORK EXPERIENCE  Senior Software Engineer  Cloud Lending Solutions -
Bangalore, Karnataka -  January 2018 to Present  Present  Senior Consultant  Oracle -  Bangalore, Karnataka -  November 2016 to
December 2017  Staff Consultant  Oracle -  Bangalore, Karnataka -  January 2014 to October 2016  Associate Consultant  Oracle -
Bangalore, Karnataka -  November 2012 to December 2013  EDUCATION  B.E in Computer Science Engineering  Adithya Institute of
Technology -  Tamil Nadu  September 2008 to June 2012  https://www.indeed.com/r/Govardhana-K/b2de315d95905b68?isid=rex-
download&ikw=download-top&co=IN https://www.indeed.com/r/Govardhana-K/b2de315d95905b68?isid=rex-download&ikw=download-top&co=IN
SKILLS  APEX. (Less than 1 year), Data Structures (3 years), FLEXCUBE (5 years), Oracle (5 years), Algorithms (3 years)  LINKS
https://www.linkedin.com/in/govardhana-k-61024944/  ADDITIONAL INFORMATION  Technical Proficiency:  Languages: Core Java, Go
Lang, Data Structures & Algorithms, Oracle PL-SQL programming, Sales Force with APEX. Tools: RADTool, Jdeveloper, NetBeans,
Eclipse, SQL developer, PL/SQL Developer, WinSCP, Putty Web Technologies: JavaScript, XML, HTML, Webservice  Operating Systems:
Linux, Windows Version control system SVN & Git-Hub Databases: Oracle Middleware: Web logic, OC4J Product FLEXCUBE: Oracle
FLEXCUBE Versions 10.x, 11.x and 12.x  https://www.linkedin.com/in/govardhana-k-61024944/',
 {'entities': [(1749, 1755, 'Companies worked at'),
   (1696, 1702, 'Companies worked at'),
   (1417, 1423, 'Companies worked at'),
   (1356, 1793, 'Skills'),
   (1209, 1215, 'Companies worked at'),
   (1136, 1248, 'Skills'),
   (928, 932, 'Graduation Year'),
   (858, 889, 'College Name'),
   (821, 856, 'Degree'),
   (787, 791, 'Graduation Year'),
   (744, 750, 'Companies worked at'),
   (722, 742, 'Designation'),
   (658, 664, 'Companies worked at'),
   (640, 656, 'Designation'),
   (574, 580, 'Companies worked at'),
   (555, 573, 'Designation'),
   (470, 493, 'Companies worked at'),
   (444, 469, 'Designation'),
   (308, 314, 'Companies worked at'),
   (234, 240, 'Companies worked at'),
   (175, 198, 'Companies worked at'),
   (93, 137, 'Email Address'),
   (39, 48, 'Location'),
   (13, 38, 'Designation'),
   (0, 12, 'Name')]})
```

```
nlp = spacy.blank('en')

def train_model(train_data):
    if 'ner' not in nlp.pipe_names:
        ner = nlp.create_pipe('ner')
        nlp.add_pipe('ner', last = True)

    for _, annotation in train_data:
        for ent in annotation['entities']:
            ner.add_label(ent[2])


    other_pipes = [pipe for pipe in nlp.pipe_names if pipe != 'ner']
    with nlp.disable_pipes(*other_pipes):  # only train NER
        optimizer = nlp.begin_training()
        for itn in range(10):
            print("Statring iteration " + str(itn))
            random.shuffle(train_data)
            losses = {}
            index = 0
            for text, annotations in train_data:
                try:
                    nlp.update(
                        [text],  # batch of texts
                        [annotations],  # batch of annotations
                        drop=0.2,  # dropout - make it harder to memorise data
                        sgd=optimizer,  # callable to update weights
                        losses=losses)
                except Exception as e:
                    pass

            print(losses)
```

```
train_model(train_data)
```

```
Statring iteration 0
{}
Statring iteration 1
{}
Statring iteration 2
{}
Statring iteration 3
{}
Statring iteration 4
{}
Statring iteration 5
{}
Statring iteration 6
{}
Statring iteration 7
{}
Statring iteration 8
{}
Statring iteration 9
{}
```

```
nlp.to_disk('nlp_model')
```

```
nlp_model = spacy.load('nlp_model')
```

```
train_data[0][0]
```

```
'Govardhana K Senior Software Engineer  Bengaluru, Karnataka, Karnataka - Email me o
n Indeed: indeed.com/r/Govardhana-K/ b2de315d95905b68  Total IT experience 5 Years 6
Months Cloud Lending Solutions INC 4 Month • Salesforce Developer Oracle 5 Years 2 M
onth • Core Java Developer Languages Core Java, Go Lang Oracle PL-SQL programming, S
ales Force Developer with APEX.  Designations & Promotions  Willing to relocate: Any
where  WORK EXPERIENCE  Senior Software Engineer  Cloud Lending Solutions -  Bangalo
re, Karnataka -  January 2018 to Present  Present  Senior Consultant  Oracle -  Bang
alore, Karnataka -  November 2016 to December 2017  Staff Consultant  Oracle -  Bang
alore, Karnataka -  January 2014 to October 2016  Associate Consultant  Oracle -  Ba
```

```
import spacy
```

```
nlp_model = spacy.load('en_core_web_sm')
```

```
doc = nlp_model(train_data[0][0])
for ent in doc.ents:
    print(f'{ent.label_.upper():{30}}- {ent.text}')
```

```
PERSON                        - Govardhana K
PERSON                        - Software Engineer
GPE                           - Bengaluru
GPE                           - Karnataka
PERSON                        - Karnataka - Email
PERSON                        - b2de315d95905b68
DATE                          - 5 Years 6
DATE                          - 4 Month
PERSON                        - PL-SQL
ORG                           - Designations & Promotions
PERSON                        - Karnataka -
DATE                          - January 2018
PERSON                        - Karnataka -
DATE                          - November 2016 to December 2017
PERSON                        - Karnataka -
DATE                          - January 2014 to October 2016
PERSON                        - Associate Consultant  Oracle -  Bangalore
PERSON                        - Karnataka -
DATE                          - November 2012 to December 2013
GPE                           - EDUCATION
ORG                           - B.E in Computer Science Engineering  Adithya Institute of Technology -
PERSON                        - Tamil Nadu
DATE                          - September 2008 to June 2012
LOC                           - APEX
DATE                          - Less than 1 year
ORG                           - Data Structures
DATE                          - 3 years
DATE                          - 5 years
ORG                           - Oracle (
DATE                          - 5 years
PERSON                        - Algorithms
DATE                          - 3 years
PERSON                        - Core Java
ORG                           - Data Structures & Algorithms
ORG                           - Oracle PL-SQL
PERSON                        - Sales Force
LOC                           - APEX
```

```
GPE                            - Jdeveloper
ORG                            - NetBeans
PRODUCT                        - Eclipse
ORG                            - SQL
ORG                            - PL/SQL Developer
PERSON                         - Putty Web Technologies
ORG                            - JavaScript
ORG                            - XML
ORG                            - HTML
ORG                            - Webservice
GPE                            - Linux
ORG                            - Windows Version
ORG                            - SVN & Git-Hub Databases
PERSON                         - OC4J Product
ORG                            - Oracle FLEXCUBE Versions
CARDINAL                       - 11.x
CARDINAL                       - 12.x
```

```
!pip install PyMuPDF
```

```
Collecting PyMuPDF
  Downloading PyMuPDF-1.24.0-cp310-none-manylinux2014_x86_64.whl (3.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.9/3.9 MB 22.3 MB/s eta 0:00:00
Collecting PyMuPDFb==1.24.0 (from PyMuPDF)
  Downloading PyMuPDFb-1.24.0-py3-none-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (30.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 30.8/30.8 MB 25.4 MB/s eta 0:00:00
Installing collected packages: PyMuPDFb, PyMuPDF
Successfully installed PyMuPDF-1.24.0 PyMuPDFb-1.24.0
```

```
import sys, fitz
fname = 'Alice Clark CV.pdf'
doc = fitz.open(fname)
text = ""
for page in doc:
    text = text + str(page.get_text())

tx = " ".join(text.split('\n'))
print(tx)
```

```
Alice Clark  AI / Machine Learning    Delhi, India Email me on Indeed  •  20+ years of experience in data handling, design, and de
```

```
doc = nlp_model(tx)
for ent in doc.ents:
    print(f'{ent.label_.upper():{30}}- {ent.text}')
```

```
PERSON                         - Alice Clark
ORG                            - AI / Machine Learning
GPE                            - Delhi
GPE                            - India
DATE                           - 20+ years
ORG                            - SQL
PERSON                         - Stored Procedures
ORG                            - Microsoft
PERSON                         - Document DB
ORG                            - SQL Azure
ORG                            - Stream Analytics
ORG                            - Power BI
GPE                            - Web Job
ORG                            - Software Engineer
ORG                            - Microsoft
GPE                            - Karnataka
DATE                           - January 2000
CARDINAL                       - 1
ORG                            - Microsoft
ORG                            - Microsoft
ORG                            - Microsoft Rewards
ORG                            - Bing
ORG                            - Microsoft Edge
FAC                            - the Xbox Store
ORG                            - Microsoft
ORG                            - Microsoft
GPE                            - US
GPE                            - Canada
GPE                            - Australia
DATE                           - weekly
TIME                           - 5 seconds to 30 minutes
ORG                            - Technology/Tools
ORG                            - Indian Institute of Technology
GPE                            - Mumbai
DATE                           - 2001
PRODUCT                        - • Quick
PRODUCT                        - • Positive
PRODUCT                        - • Supervised
```

```
import sys, fitz
fname = 'Smith Resume.pdf'
doc = fitz.open(fname)
text = ""
for page in doc:
    text = text + str(page.get_text())

tx = " ".join(text.split('\n'))
print(tx)
```

```
 Michael Smith  BI / Big Data/ Azure  Manchester, UK- Email me on Indeed: indeed.com/r/falicent/140749dace5dc26f    10+ years of Ex
```

```
doc = nlp_model(tx)
for ent in doc.ents:
    print(f'{ent.label_.upper():{30}}- {ent.text}')
```

```
PERSON                        - Michael Smith
ORG                           - BI / Big Data/
ORG                           - Manchester
ORG                           - Designing, Development, Administration
GPE                           - Client
ORG                           - Server  Technologies
ORG                           - Applications
ORG                           - SQL
PERSON                        - Stored Procedures
ORG                           - Microsoft
PERSON                        - Document DB
ORG                           - SQL  Azure
NORP                          - StreamAnalytics
ORG                           - Power BI
GPE                           - Web Job
ORG                           - U-SQL
ORG                           - Microsoft - Manchester
GPE                           - UK
DATE                          - December 2015
CARDINAL                      - 1
ORG                           - Microsoft
ORG                           - Microsoft
ORG                           - Microsoft Rewards
ORG                           - Bing
ORG                           - Microsoft Edge
PRODUCT                       - Xbox Store
ORG                           - the Microsoft Store
ORG                           - Microsoft
NORP                          - Rewards
GPE                           - US
GPE                           - Canada
GPE                           - Australia
DATE                          - weekly
TIME                          - 5 seconds to 30 minutes
ORG                           - Technology/Tools
ORG                           - Power BI
ORG                           - Responsibilities  Created
ORG                           - Created Power BI
DATE                          - weekly
CARDINAL                      - 10
CARDINAL                      - 2
ORG                           - Microsoft
ORG                           - Microsoft
ORG                           - Microsoft Rewards
ORG                           - Bing
ORG                           - Microsoft Edge
PRODUCT                       - Xbox Store
ORG                           - the Microsoft Store
ORG                           - Microsoft
CARDINAL                      - 20  million
DATE                          - daily
GPE                           - US
GPE                           - Canada
GPE                           - Australia
ORG                           - Technology/Tools
PERSON                        - Cosmos
ORG                           - Microsoft
CARDINAL                      - #
```

```
import spacy
import re
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.neural_network import MLPClassifier
from sklearn.pipeline import make_pipeline
import csv

# Load spaCy NER model
nlp = spacy.load("en_core_web_sm")
```

```
# Sample resumes
resumes = [
    "John Doe\nEmail: john.doe@example.com\nPhone: 123-456-7890\nSkills: Python, Machine Learning",
    "Jane Smith\nEmail: jane.smith@example.com\nPhone: 987-654-3210\nExperience: Data Analyst",
    "Alice Johnson\nEmail: alice.johnson@example.com\nPhone: 111-222-3333\nSkills: Java, C++"
]

# Initialize lists to store extracted information
names = []
emails = []

# Extract names and emails using spaCy
for resume in resumes:
    doc = nlp(resume)

    name = None
    email = None

    for ent in doc.ents:
        if ent.label_ == "PERSON":
            name = ent.text
        elif ent.label_ == "EMAIL":
            email = ent.text

    names.append(name)
    emails.append(email)

# Prepare data for ANN model
data = pd.DataFrame({
    "Name": names,
    "Email": emails
})

X = data["Name"].fillna("").astype(str) + " " + data["Email"].fillna("").astype(str)
y = data["Email"].notnull().astype(int)

# Create pipeline with CountVectorizer and MLPClassifier
pipeline = make_pipeline(
    CountVectorizer(),
    MLPClassifier(hidden_layer_sizes=(50,), max_iter=100, alpha=1e-4, solver="sgd", verbose=10, random_state=1, learning_rate_init=.1)
)

# Train ANN model
pipeline.fit(X, y)

# Test the model
test_resumes = [
    "John Doe\nEmail: john.doe@example.com\nPhone: 123-456-7890\nSkills: Python, Machine Learning",
    "Jane Smith\nEmail: jane.smith@example.com\nPhone: 987-654-3210\nExperience: Data Analyst",
    "Alice Johnson\nEmail: alice.johnson@example.com\nPhone: 111-222-3333\nSkills: Java, C++"
]

test_names = []
test_emails = []

for resume in test_resumes:
    doc = nlp(resume)

    name = None
    email = None

    for ent in doc.ents:
        if ent.label_ == "PERSON":
            name = ent.text
        elif ent.label_ == "EMAIL":
            email = ent.text

    test_names.append(name)
    test_emails.append(email)

test_data = pd.DataFrame({
    "Name": test_names,
    "Email": test_emails
})

test_X = test_data["Name"].fillna("").astype(str) + " " + test_data["Email"].fillna("").astype(str)

predictions = pipeline.predict(test_X)

# Print test results
for i, resume in enumerate(test_resumes):
    print(f"Resume:\n{resume}\nPredicted Email: {'Yes' if predictions[i] == 1 else 'No'}\n")
```

```
print(f'Resume:\n{resume}\nPredicted Email: {"Yes" if predictions[i] == 1 else "No"}\n')
```

```
Iteration 1, loss = 0.87070568
Iteration 2, loss = 0.61645482
Iteration 3, loss = 0.38656675
Iteration 4, loss = 0.22226708
Iteration 5, loss = 0.12218620
Iteration 6, loss = 0.06666814
Iteration 7, loss = 0.03690494
Iteration 8, loss = 0.02104933
Iteration 9, loss = 0.01240359
Iteration 10, loss = 0.00759648
Iteration 11, loss = 0.00484842
Iteration 12, loss = 0.00323108
Iteration 13, loss = 0.00225465
Iteration 14, loss = 0.00164850
Iteration 15, loss = 0.00126129
Iteration 16, loss = 0.00100836
Iteration 17, loss = 0.00083918
Iteration 18, loss = 0.00072350
Iteration 19, loss = 0.00064280
Iteration 20, loss = 0.00058532
Iteration 21, loss = 0.00054374
Iteration 22, loss = 0.00051327
Iteration 23, loss = 0.00049063
Iteration 24, loss = 0.00047360
Iteration 25, loss = 0.00046064
Iteration 26, loss = 0.00045067
Iteration 27, loss = 0.00044294
Iteration 28, loss = 0.00043689
Iteration 29, loss = 0.00043211
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Resume:
John Doe
Email: john.doe@example.com
Phone: 123-456-7890
Skills: Python, Machine Learning
Predicted Email: No

Resume:
Jane Smith
Email: jane.smith@example.com
Phone: 987-654-3210
Experience: Data Analyst
Predicted Email: No

Resume:
Alice Johnson
Email: alice.johnson@example.com
Phone: 111-222-3333
Skills: Java, C++
Predicted Email: No
```