Load Spacy Model

In [1]:
```python
import spacy
nlp = spacy.load("en_core_web_sm")
import json
```

In [64]:
```python
import random
from spacy.util import minibatch, compounding
from spacy.training import Example
import pickle
from pathlib import Path
import warnings
warnings.filterwarnings('ignore')
import numpy as np
```

In [8]:
```python
res_file = open("Entity Recognition in Resumes.json",'r',encoding='utf-8')
json_str = []
for line in res_file:
    json_str.append(line)
```

In [27]:
```python
json.loads(json_str[1])
```

Out[27]:

```
{'content': 'Afreen Jamadar\nActive member of IIIT Committee in Third year\n\nSangli, Maharashtra - Email me on Indeed: indeed.com/r/Afreen-Jamadar/8baf379b705e37c6\n\nI wish to use my knowledge, skills and conceptual understanding to create excellent team\nenvironments and work consistently achieving organization objectives believes in taking initiative\nand work to excellence in my work.\n\nWORK EXPERIENCE\n\nActive member of IIIT Committee in Third year\n\nCisco Networking -  Kanpur, Uttar Pradesh\n\norganized by Techkriti IIT Kanpur and Azure Skynet.\nPERSONALLITY TRAITS:\n• Quick learning ability\n• hard working\n\nEDUCATION\n\nPG-DAC\n\nCDAC ACTS\n\n2017\n\nBachelor of Engg in Information Technology\n\nShivaji University Kolhapur -  Kolhapur, Maharashtra\n\n2016\n\nSKILLS\n\nDatabase (Less than 1 year), HTML (Less than 1 year), Linux. (Less than 1 year), MICROSOFT\nACCESS (Less than 1 year), MICROSOFT WINDOWS (Less than 1 year)\n\nADDITIONAL INFORMATION\n\nTECHNICAL SKILLS:\n\n• Programming Languages: C, C++, Java, .net, php.\n• Web Designing: HTML, XML\n• Operating Systems: Windows […] Windows Server 2003, Linux.\n• Database: MS Access, MS SQL Server 2008, Oracle 10g, MySql.\n\nhttps://www.indeed.com/r/Afreen-Jamadar/8baf379b705e37c6?isid=rex-download&ikw=download-top&co=IN',
 'annotation': [{'label': ['Email Address'],
   'points': [{'start': 1155,
     'end': 1198,
     'text': 'indeed.com/r/Afreen-Jamadar/8baf379b705e37c6'}]},
  {'label': ['Skills'],
   'points': [{'start': 743,
     'end': 1140,
     'text': 'Database (Less than 1 year), HTML (Less than 1 year), Linux. (Less than 1 year), MICROSOFT\nACCESS (Less than 1 year), MICROSOFT WINDOWS (Less than 1 year)\n\nADDITIONAL INFORMATION\n\nTECHNICAL SKILLS:\n\n• Programming Languages: C, C++, Java, .net, php.\n• Web Designing: HTML, XML\n• Operating Systems: Windows […] Windows Server 2003, Linux.\n• Database: MS Access, MS SQL Server 2008, Oracle 10g, MySql.'}]},
  {'label': ['Graduation Year'],
   'points': [{'start': 729, 'end': 732, 'text': '2016'}]},
  {'label': ['College Name'],
   'points': [{'start': 675,
     'end': 702,
     'text': 'Shivaji University Kolhapur '}]},
  {'label': ['Degree'],
   'points': [{'start': 631,
     'end': 672,
     'text': 'Bachelor of Engg in Information Technology'}]},
  {'label': ['Graduation Year'],
   'points': [{'start': 625, 'end': 629, 'text': '2017\n'}]},
  {'label': ['College Name'],
   'points': [{'start': 614, 'end': 622, 'text': 'CDAC ACTS'}]},
  {'label': ['Degree'],
   'points': [{'start': 606, 'end': 611, 'text': 'PG-DAC'}]},
  {'label': ['Email Address'],
   'points': [{'start': 104,
     'end': 147,
     'text': 'indeed.com/r/Afreen-Jamadar/8baf379b705e37c6'}]},
  {'label': ['Location'],
   'points': [{'start': 62, 'end': 67, 'text': 'Sangli'}]},
  {'label': ['Name'],
   'points': [{'start': 0, 'end': 13, 'text': 'Afreen Jamadar'}]}],
 'extras': None}
```


Prepare Data For Spacy

In [ ]:
```python
# This function to remove overlaping entities
def filter_anotation(anLst):
    ss_n = []
    prev_ij = []

    for i,j,s in anLst:
        if prev_ij == []:
            prev_ij.append([i,j])
            ss_n.append((i,j,s))
        else:
            not_pre = True
            for ii in prev_ij:
                a = ii[0]
                b = ii[1]
                if ((i >= a and i <b) or (j >= a and j <b)) or ((a >= i and a <j) or
                    not_pre = False

            if [i,j] not in prev_ij and not_pre:
                prev_ij.append([i,j])
                ss_n.append((i,j,s))
    return ss_n
```

In [39]:
```python
sk_set = []
lr = []
OL = []
labels = []
for json_entr in json_str:
    d = json.loads(json_entr)
    cont = " ".join(d['content'].split("\n"))
    cont  = " ".join([sd.text.lower() for sd in nlp.tokenizer(cont) if not sd.is_punct
    ets = []
    for ano in d['annotation']:
        if len(ano['label']) != 0:
            lbl = "_".join(ano['label'][0].lower().split(" "))

            txt =  " ".join(ano['points'][0]['text'].split('\n'))
            txt = [sd.text for sd in nlp.tokenizer(" ".join(txt.split(","))) if not sd
            txt = [t.text.lower() for t in nlp.tokenizer(ano['points'][0]['text']) if

            for ww in txt:
                s = cont.find(ww)
                e = s + len(ww)
                if s!=-1:
                    OL.append([ww,lbl])
                    if e-s >2:
                        tpl = (s,e,lbl)
                        ets.append(tpl)
    ets = filter_anotation(ets)
    ets = list(set(ets))
    if len(ets) != 0:
        lr.append((cont,{'entities':ets}))
```

In [65]:
```python
n = len(lr)
test_per = 0.2
train_data = lr[:int(n*(1-test_per))]
test_data = lr[-1*int(n*test_per):]
```

In [66]:
```python
len(train_data),len(test_data)
```

Out[66]: (176, 44)

In [122...
```python
test_data[5]
```

Out[122...
```
('nida khan tech support executive teleperformance for microsoft   jaipur rajasthan email me on indeed indeed.com/r/nida-khan/6c9160696f57efd8   to be an integral part of the organization and enhance my knowledge to utilize it in a productive manner for the growth of the company and the global   industrial training   bhel heep haridwar on cnc system&amp plc programming   work experience   tech support executive   teleperformance for microsoft   september 2017 to present   process 21 months of experience in adfc as phone banker   education   bachelor of technology in electronics communication engg   gnit institute of technology   lucknow uttar pradesh   2008 to 2012   class xii   u.p. board   bareilly uttar pradesh   2007   class x   u.p. board   bareilly uttar pradesh   2005   skills   microsoft office excel cisco c language cbs 4 years   https://www.indeed.com/r/nida-khan/6c9160696f57efd8?isid=rex-download&ikw=download-top&co=in',
 {'entities': [(557, 567, 'college_name'),
   (805, 811, 'skills'),
   (835, 838, 'skills'),
   (23, 32, 'designation'),
   (818, 823, 'skills'),
   (841, 846, 'skills'),
   (583, 596, 'degree'),
   (10, 14, 'designation'),
   (33, 48, 'companies_worked_at'),
   (571, 582, 'degree'),
   (609, 618, 'college_name'),
   (5, 9, 'name'),
   (604, 608, 'college_name'),
   (65, 71, 'location'),
   (597, 601, 'degree'),
   (545, 553, 'degree'),
   (0, 4, 'name'),
   (812, 817, 'skills'),
   (53, 62, 'skills'),
   (826, 834, 'skills'),
   (15, 22, 'designation')]})
```

Prepare new NER Spacy Pipeline

In [73]:
```python
test_nlp = spacy.blank('en')
ner = test_nlp.create_pipe('ner')
test_nlp.add_pipe('ner')
```

Out[73]: <spacy.pipeline.ner.EntityRecognizer at 0x1ce32143100>

In [74]:
```python
for _,et in train_data:
    for sss in et['entities']:
        lbl = sss[-1]
        ner.add_label(lbl)
```

In [75]:
```python
optimizer = test_nlp.begin_training()
```

In [76]:
```python
p= 0
batch_size = 10
for _ in range(400):
    losses = {}
    random.shuffle(train_data)

    for i in range(0,len(train_data),batch_size):
        batch = []
        for raw_text, entity_offsets in train_data[i:i+batch_size]:
            doc = test_nlp.make_doc(raw_text)
            example = Example.from_dict(doc, entity_offsets)
            batch.append(example)
        test_nlp.update(batch, sgd=optimizer, drop=0.2,
                        losses=losses)
    p = p+1
    print('Losses', losses,p)
```

```
Losses {'ner': 46402.300721591106} 1
Losses {'ner': 10557.318391839042} 2
Losses {'ner': 9932.518139360473} 3
Losses {'ner': 9492.379184351303} 4
Losses {'ner': 9227.639436833793} 5
Losses {'ner': 8585.594196146354} 6
Losses {'ner': 8364.281489412882} 7
Losses {'ner': 8071.230324698612} 8
Losses {'ner': 7713.988535086624} 9
Losses {'ner': 7382.056466132868} 10
Losses {'ner': 7188.287369651953} 11
Losses {'ner': 7670.2755635991925} 12
Losses {'ner': 6822.523830083024} 13
Losses {'ner': 6913.090414646082} 14
Losses {'ner': 6649.963975582694} 15
Losses {'ner': 6357.061282712617} 16
Losses {'ner': 6123.230789102847} 17
Losses {'ner': 5899.534422313212} 18
Losses {'ner': 5749.906565129553} 19
Losses {'ner': 5615.244502707152} 20
Losses {'ner': 5486.020955686021} 21
Losses {'ner': 5547.594596983616} 22
Losses {'ner': 5420.022360139759} 23
Losses {'ner': 5277.461785773281} 24
Losses {'ner': 5059.464160461928} 25
Losses {'ner': 4989.017748712096} 26
Losses {'ner': 4951.846625936669} 27
Losses {'ner': 4686.961060921385} 28
Losses {'ner': 4502.228097052488} 29
Losses {'ner': 4336.901949367602} 30
Losses {'ner': 4251.145652488165} 31
Losses {'ner': 4124.935175527164} 32
Losses {'ner': 4150.835382136051} 33
Losses {'ner': 3959.538672475035} 34
Losses {'ner': 3817.158955209845} 35
Losses {'ner': 3957.7988065737154} 36
Losses {'ner': 3734.9092246287764} 37
Losses {'ner': 3751.214244696693} 38
Losses {'ner': 3784.1908103333844} 39
Losses {'ner': 3602.9356923519445} 40
Losses {'ner': 3600.2534318690487} 41
Losses {'ner': 3391.0937287856686} 42
Losses {'ner': 3345.985771202417} 43
Losses {'ner': 3190.4114341125232} 44
Losses {'ner': 3216.039958620316} 45
Losses {'ner': 3123.2353360005254} 46
Losses {'ner': 3201.2911407861866} 47
Losses {'ner': 3053.791221569834} 48
Losses {'ner': 3075.1029517589654} 49
Losses {'ner': 2908.832655436556} 50
Losses {'ner': 2756.008622227073} 51
Losses {'ner': 2954.4844572045795} 52
Losses {'ner': 2768.8776905445748} 53
Losses {'ner': 2675.2310878460303} 54
Losses {'ner': 2863.5551835217047} 55
Losses {'ner': 2603.3955423013404} 56
Losses {'ner': 2556.637229622582} 57
Losses {'ner': 2647.57532987699} 58
Losses {'ner': 2529.1997655211007} 59
Losses {'ner': 2361.30100616005} 60
Losses {'ner': 2373.211701868512} 61
Losses {'ner': 2403.4681945521907} 62
Losses {'ner': 2300.6488305800517} 63
Losses {'ner': 2285.6215197989554} 64
Losses {'ner': 2257.3149460037484} 65
Losses {'ner': 2131.5496743891877} 66
Losses {'ner': 2272.7087869275447} 67
Losses {'ner': 2270.7329048946685} 68
Losses {'ner': 2115.9652315475955} 69
Losses {'ner': 2156.09518774916} 70
Losses {'ner': 2035.4483890565919} 71
Losses {'ner': 2130.1295984797216} 72
Losses {'ner': 2125.743859906782} 73
Losses {'ner': 1954.3848484906289} 74
Losses {'ner': 2053.700672208222} 75
Losses {'ner': 2014.6491004694517} 76
Losses {'ner': 1940.729056115161} 77
Losses {'ner': 1890.1125432021606} 78
Losses {'ner': 1958.2404552699816} 79
Losses {'ner': 1927.7067912630316} 80
Losses {'ner': 1799.1478379073965} 81
Losses {'ner': 1753.1152739485187} 82
Losses {'ner': 1754.9547468977999} 83
Losses {'ner': 1739.298770009081} 84
Losses {'ner': 1749.8787464589623} 85
Losses {'ner': 1761.8141920283738} 86
Losses {'ner': 1767.32412326544} 87
Losses {'ner': 1703.5520471299726} 88
Losses {'ner': 1626.2149936481542} 89
Losses {'ner': 1559.2797372801701} 90
Losses {'ner': 1586.047033943185} 91
Losses {'ner': 1607.6696640455618} 92
Losses {'ner': 1555.3312780155136} 93
Losses {'ner': 1576.1195396567834} 94
Losses {'ner': 1508.9357493250225} 95
Losses {'ner': 1549.580373885588} 96
Losses {'ner': 1536.4390823053623} 97
Losses {'ner': 1544.816679517119} 98
Losses {'ner': 1544.222546883945} 99
Losses {'ner': 1405.422695789537} 100
Losses {'ner': 1577.7306958014801} 101
Losses {'ner': 1458.1733710699175} 102
Losses {'ner': 1478.0433533532969} 103
Losses {'ner': 1371.7387033202872} 104
Losses {'ner': 1325.1772316561505} 105
Losses {'ner': 1456.8639858642955} 106
Losses {'ner': 1391.5980159123208} 107
Losses {'ner': 1373.4166883445391} 108
Losses {'ner': 1341.0646918090056} 109
Losses {'ner': 1314.519222845266} 110
Losses {'ner': 1373.0261386472318} 111
Losses {'ner': 1312.9057371434637} 112
Losses {'ner': 1274.7633859511905} 113
Losses {'ner': 1261.9832082987714} 114
Losses {'ner': 1299.8194129673097} 115
Losses {'ner': 1257.4787122656878} 116
Losses {'ner': 1224.7097145723155} 117
Losses {'ner': 1276.1775038840913} 118
Losses {'ner': 1224.4928234452059} 119
Losses {'ner': 1243.0218680557605} 120
Losses {'ner': 1298.971294011752} 121
Losses {'ner': 1180.3953815052437} 122
Losses {'ner': 1236.5827046353947} 123
Losses {'ner': 1151.489303146719} 124
Losses {'ner': 1205.2622632545176} 125
Losses {'ner': 1157.4909537816666} 126
Losses {'ner': 1202.3440119238508} 127
Losses {'ner': 1164.7811118317911} 128
Losses {'ner': 1097.889779996019} 129
Losses {'ner': 1116.2302076488188} 130
Losses {'ner': 1077.3429598536914} 131
Losses {'ner': 1195.488530728312} 132
Losses {'ner': 1131.9301101111182} 133
Losses {'ner': 1160.7777886733118} 134
Losses {'ner': 1097.0762145346937} 135
Losses {'ner': 1121.3005479922235} 136
Losses {'ner': 1120.6587174204546} 137
Losses {'ner': 1080.2403224039374} 138
Losses {'ner': 1036.63449644358} 139
Losses {'ner': 1037.80133165584} 140
Losses {'ner': 1048.4112646053056} 141
Losses {'ner': 1100.3797685348752} 142
Losses {'ner': 1112.0638194933147} 143
Losses {'ner': 1020.3873648741534} 144
Losses {'ner': 989.5830560477315} 145
Losses {'ner': 1001.4091155965242} 146
Losses {'ner': 1045.762880429219} 147
Losses {'ner': 965.1317267672057} 148
Losses {'ner': 1057.1247512664868} 149
Losses {'ner': 1011.1264457702305} 150
Losses {'ner': 932.3229356239386} 151
Losses {'ner': 955.815118598389} 152
Losses {'ner': 1043.5570813797262} 153
Losses {'ner': 1035.8809156222087} 154
Losses {'ner': 932.5176407189613} 155
Losses {'ner': 969.2545904695291} 156
Losses {'ner': 1003.3612323243591} 157
Losses {'ner': 1053.4688821571535} 158
Losses {'ner': 978.9967551349748} 159
Losses {'ner': 955.5550894733035} 160
Losses {'ner': 965.3584358398608} 161
Losses {'ner': 906.1035331700438} 162
Losses {'ner': 906.3642183286062} 163
Losses {'ner': 912.8891393758037} 164
Losses {'ner': 933.6899149593785} 165
Losses {'ner': 952.4642284716846} 166
Losses {'ner': 878.1956783196546} 167
Losses {'ner': 866.7953300010589} 168
Losses {'ner': 903.8695877313069} 169
Losses {'ner': 907.1139724664157} 170
Losses {'ner': 927.1603594238427} 171
Losses {'ner': 938.1530739184678} 172
Losses {'ner': 868.6888070994642} 173
Losses {'ner': 902.9024050946215} 174
Losses {'ner': 850.2017577842655} 175
Losses {'ner': 919.3598736421982} 176
Losses {'ner': 854.4713258265682} 177
Losses {'ner': 870.8831508327096} 178
Losses {'ner': 890.0836313261789} 179
Losses {'ner': 818.3027558887636} 180
Losses {'ner': 862.8987765745243} 181
Losses {'ner': 775.3355301463047} 182
Losses {'ner': 798.0496309602015} 183
Losses {'ner': 929.1356738313698} 184
Losses {'ner': 767.0282259750763} 185
Losses {'ner': 840.5356822167284} 186
Losses {'ner': 823.1625307862419} 187
Losses {'ner': 803.2060213264426} 188
Losses {'ner': 790.997013246887} 189
Losses {'ner': 829.4973486404879} 190
Losses {'ner': 784.660043233024} 191
Losses {'ner': 836.7515615741925} 192
Losses {'ner': 734.4640396458503} 193
Losses {'ner': 788.9095847362928} 194
Losses {'ner': 815.5122058325295} 195
Losses {'ner': 776.7406523936859} 196
Losses {'ner': 760.4637943246842} 197
Losses {'ner': 709.0585303629089} 198
Losses {'ner': 776.3341468338848} 199
Losses {'ner': 758.3760693461288} 200
Losses {'ner': 813.2886940400381} 201
Losses {'ner': 709.2239924170082} 202
Losses {'ner': 755.0703505605254} 203
Losses {'ner': 713.9413073415888} 204
Losses {'ner': 772.7226563711986} 205
Losses {'ner': 709.1484140817471} 206
Losses {'ner': 783.7360153926861} 207
Losses {'ner': 738.1328327160869} 208
Losses {'ner': 695.5804873213362} 209
Losses {'ner': 754.9349057644607} 210
Losses {'ner': 721.7500251675785} 211
Losses {'ner': 709.5764084463799} 212
Losses {'ner': 662.5434128717151} 213
Losses {'ner': 722.7581411449144} 214
Losses {'ner': 723.4956005836193} 215
Losses {'ner': 675.2158124895235} 216
Losses {'ner': 691.2770552580599} 217
Losses {'ner': 804.7281479980868} 218
Losses {'ner': 725.6503929908797} 219
Losses {'ner': 746.5868445308811} 220
Losses {'ner': 678.6879661030772} 221
Losses {'ner': 719.409944490706} 222
Losses {'ner': 642.734251948451} 223
Losses {'ner': 744.6949780019154} 224
Losses {'ner': 709.9120589267825} 225
Losses {'ner': 699.0537855876871} 226
Losses {'ner': 714.8432278457489} 227
Losses {'ner': 698.3932113088287} 228
Losses {'ner': 671.0344483882427} 229
Losses {'ner': 680.4467061715659} 230
Losses {'ner': 660.3281443250664} 231
Losses {'ner': 651.5837940127758} 232
Losses {'ner': 613.2484442930512} 233
Losses {'ner': 712.2924743968268} 234
Losses {'ner': 669.2309881658089} 235
Losses {'ner': 674.7401082176979} 236
Losses {'ner': 715.9757620440179} 237
Losses {'ner': 637.3145963191317} 238
Losses {'ner': 677.1958763114419} 239
Losses {'ner': 662.605094666964} 240
Losses {'ner': 687.5041447597362} 241
Losses {'ner': 676.5560402363747} 242
Losses {'ner': 678.9234597920187} 243
Losses {'ner': 657.0290729803414} 244
Losses {'ner': 654.5709867799893} 245
Losses {'ner': 682.5800818916783} 246
Losses {'ner': 644.9726834773647} 247
Losses {'ner': 693.7598501458291} 248
Losses {'ner': 616.606545782683} 249
Losses {'ner': 663.003762915844} 250
Losses {'ner': 664.2980388597967} 251
Losses {'ner': 612.5613755632696} 252
Losses {'ner': 607.5666388753579} 253
Losses {'ner': 614.3819160653227} 254
Losses {'ner': 680.4578801485493} 255
Losses {'ner': 654.295466211204} 256
Losses {'ner': 622.2378492235916} 257
Losses {'ner': 598.6588177680135} 258
Losses {'ner': 596.9923209001097} 259
Losses {'ner': 610.7869551316826} 260
Losses {'ner': 588.3633202147097} 261
Losses {'ner': 608.0282748318791} 262
Losses {'ner': 592.2271013931793} 263
Losses {'ner': 603.984789006642} 264
Losses {'ner': 585.4997572866998} 265
Losses {'ner': 612.5697196018202} 266
Losses {'ner': 616.8510022990605} 267
Losses {'ner': 642.2040766661423} 268
Losses {'ner': 569.642729910354} 269
Losses {'ner': 595.0667962088821} 270
Losses {'ner': 575.8065519776836} 271
Losses {'ner': 607.0476867319499} 272
Losses {'ner': 562.7032479789348} 273
Losses {'ner': 629.1172263632866} 274
Losses {'ner': 585.2764085696994} 275
Losses {'ner': 556.1243994406418} 276
Losses {'ner': 627.7231662415423} 277
Losses {'ner': 567.6130140359425} 278
Losses {'ner': 604.7901630324122} 279
Losses {'ner': 610.52531934798} 280
Losses {'ner': 535.4720199748236} 281
Losses {'ner': 589.6663807762578} 282
Losses {'ner': 558.5620032651282} 283
Losses {'ner': 584.391148361348} 284
Losses {'ner': 573.2225035003042} 285
Losses {'ner': 503.339835244572} 286
Losses {'ner': 590.0280124842376} 287
Losses {'ner': 617.0222077740203} 288
Losses {'ner': 539.8775108856311} 289
Losses {'ner': 563.4458216092712} 290
Losses {'ner': 608.1840489007606} 291
Losses {'ner': 545.5743850606826} 292
Losses {'ner': 550.8324575361366} 293
Losses {'ner': 567.5277350792637} 294
Losses {'ner': 547.1390413443696} 295
Losses {'ner': 508.01665435445113} 296
Losses {'ner': 547.1536059713535} 297
Losses {'ner': 545.585877889318} 298
Losses {'ner': 505.5754896868805} 299
Losses {'ner': 553.0659022818752} 300
Losses {'ner': 573.8047278990764} 301
Losses {'ner': 528.0661004985817} 302
Losses {'ner': 569.4043208767296} 303
Losses {'ner': 561.1258197516564} 304
Losses {'ner': 535.6719963525998} 305
Losses {'ner': 600.3892559785301} 306
Losses {'ner': 539.663824154836} 307
Losses {'ner': 535.2079903699689} 308
Losses {'ner': 494.5844908844499} 309
Losses {'ner': 491.8143009003851} 310
Losses {'ner': 533.1150980943836} 311
Losses {'ner': 502.94589750407715} 312
Losses {'ner': 538.8969563566644} 313
Losses {'ner': 546.9246099112776} 314
Losses {'ner': 510.7793712659843} 315
Losses {'ner': 542.1345643289399} 316
Losses {'ner': 564.6663083671546} 317
Losses {'ner': 496.7803723886194} 318
Losses {'ner': 516.112001825409} 319
Losses {'ner': 521.8357357708514} 320
Losses {'ner': 481.0801865782095} 321
Losses {'ner': 496.28534899529825} 322
Losses {'ner': 517.0796957461639} 323
Losses {'ner': 526.0641332280674} 324
Losses {'ner': 480.9685033386075} 325
Losses {'ner': 515.2528971253364} 326
Losses {'ner': 517.048044835583} 327
Losses {'ner': 508.5164098824878} 328
Losses {'ner': 523.6569449131772} 329
Losses {'ner': 479.36448061302315} 330
Losses {'ner': 524.5174263811396} 331
Losses {'ner': 519.2384888196616} 332
Losses {'ner': 504.7758598666717} 333
Losses {'ner': 491.79446843885364} 334
Losses {'ner': 485.35239124102264} 335
Losses {'ner': 508.0381880742738} 336
Losses {'ner': 522.6723596414365} 337
Losses {'ner': 498.5628446760345} 338
Losses {'ner': 479.61972200511843} 339
Losses {'ner': 507.7648057004282} 340
Losses {'ner': 461.371396107901} 341
Losses {'ner': 489.1225330437744} 342
Losses {'ner': 479.71763735728604} 343
Losses {'ner': 487.1978412545675} 344
Losses {'ner': 466.8047279428185} 345
Losses {'ner': 498.34772061075176} 346
Losses {'ner': 534.3113276112202} 347
Losses {'ner': 451.6555747028077} 348
Losses {'ner': 456.4509239348979} 349
Losses {'ner': 474.17574654368946} 350
Losses {'ner': 496.57308925762067} 351
Losses {'ner': 490.58667686494545} 352
Losses {'ner': 468.67984741011617} 353
Losses {'ner': 508.63449568960726} 354
Losses {'ner': 462.0671891355002} 355
Losses {'ner': 430.6458487250742} 356
Losses {'ner': 475.74799631241467} 357
Losses {'ner': 466.5523595943228} 358
Losses {'ner': 469.1077882045793} 359
Losses {'ner': 503.1937994453522} 360
Losses {'ner': 443.99639940361607} 361
Losses {'ner': 462.66256006298073} 362
Losses {'ner': 470.67233362624097} 363
Losses {'ner': 476.98611628671676} 364
Losses {'ner': 463.1809195808796} 365
Losses {'ner': 436.1901326378014} 366
Losses {'ner': 513.9729220478475} 367
Losses {'ner': 529.0202491144659} 368
Losses {'ner': 503.7168857491975} 369
Losses {'ner': 467.35742587388415} 370
Losses {'ner': 449.5822657783123} 371
Losses {'ner': 459.1964204951069} 372
Losses {'ner': 473.50408217657747} 373
Losses {'ner': 465.2796596093117} 374
Losses {'ner': 446.54135501822526} 375
Losses {'ner': 493.9017059205252} 376
Losses {'ner': 472.87600958256627} 377
Losses {'ner': 485.8890070085252} 378
Losses {'ner': 455.99413415285125} 379
Losses {'ner': 419.96177133892434} 380
Losses {'ner': 478.0161709405556} 381
Losses {'ner': 475.8187421535969} 382
Losses {'ner': 479.3634138154863} 383
Losses {'ner': 440.6539183082341} 384
Losses {'ner': 453.7635854837731} 385
Losses {'ner': 438.3324992690649} 386
Losses {'ner': 447.17003788294267} 387
Losses {'ner': 444.37344550202954} 388
Losses {'ner': 439.81561132539684} 389
Losses {'ner': 452.27242189524827} 390
Losses {'ner': 448.3863577141597} 391
Losses {'ner': 430.5546791073797} 392
Losses {'ner': 446.1147196122607} 393
Losses {'ner': 481.5016813356551} 394
Losses {'ner': 421.7115193712588} 395
Losses {'ner': 453.9785621669082} 396
Losses {'ner': 426.5918517172909} 397
Losses {'ner': 415.6357418652359} 398
Losses {'ner': 438.3293434906663} 399
Losses {'ner': 440.44717931534655} 400
```

Test Accuracy

In [107...
```python
train_all_accuracy = []
for resume_data in train_data:
    content = resume_data[0]
    anot = resume_data[1]['entities']
    words = [content[an[0]:an[1]] for an in anot]
    nlp_content = test_nlp(content)

    found_ents = [e.text for e in nlp_content.ents]
    correct_found = [e for e in found_ents if e in words]
    eta = len(correct_found)/len(words)
    train_all_accuracy.append(eta)
acc = np.mean(train_all_accuracy)
print("Train Accuracy:",acc*100)
```

Train Accuracy: 97.52532454504941

In [106...
```python
dd = test_nlp(train_data[10][0])
spacy.displacy.render(dd,style='ent')
```

kavya name u. network designation ops designation associate designation accenture companies_worked_at bengaluru location karnataka email me on indeed indeed.com/r/kavya-u/049577580b3814e6 seeking for opportunities to learn and grow in electronics domain work experience network ops associate accenture bengaluru karnataka december 2017 to present exposure provisioning of different types of network speeds for multiple clients use of mux designing for logical cross connect and physical patching we co- ordinate with various ops and field engineers to do the connections at the physical level 2 organisation quadgen wireless engineering degree services pvt ltd. bangalore position network engineer experience 1 year exposure ran radio access network engineer new site build nsb 2 organisation manipal college_name dot net pvt ltd. manipal position intern experience 1 year exposure module coding skills and verification using verilog skills hdl skills worked on linux o.s. understanding skills of spi i2c protocols compilation using altera quartus simulation using modelsim report preparation and documentation education college_name little college_name rock college_name indian college_name school college_name 2007 master degree of science degree in technology college_name in technology school of information college_name sciences college_name vlsi design https://www.indeed.com/r/kavya-u/049577580b3814e6?isid=rex-download&ikw=download-top&co=in manipal academy college_name of higher college_name education bachelor degree of engineering in engineering srinivas college_name institute college_name of technology mangalore karnataka electronics and communication visvesvaraya college_name technological college_name university college_name vidyodaya college_name p.u. college college_name udipi karnataka skills coding less than 1 year hdl less than 1 year microsoft skills office skills less than 1 year ms office skills less than 1 year uart less than 1 year additional skills information technical skills verilog hdl knowledge skills of rtl skills coding fsm skills based skills designs skills understanding of uart amba skills protocol platforms microsoft office libreoffice skills

In [108...
```python
all_accuracy = []
for resume_data in test_data:
    content = resume_data[0]
    anot = resume_data[1]['entities']
    words = [content[an[0]:an[1]] for an in anot]
    nlp_content = test_nlp(content)

    found_ents = [e.text for e in nlp_content.ents]
    correct_found = [e for e in found_ents if e in words]
    eta = len(correct_found)/len(words)
    all_accuracy.append(eta)
acc = np.mean(all_accuracy)
print("Test Accuracy:",acc*100)
```

Test Accuracy: 53.367343091299546

In [115...
```python
dd = test_nlp(test_data[38][0])
spacy.displacy.render(dd,style='ent')
```

ramkrishan name bhatt name python skills developer designation bengaluru location karnataka karnataka email me on indeed indeed.com/r/ramkrishan-bhatt/ da07dc6d058dfc64 work experience python skills developer designation microsoft bengaluru karnataka june 2017 to november 2017 career summary 7.5 years it related employment as a software developer expert in python web development familiar with django flask web2py tensorflow jupyter notebook machine learning and artificial intelligence played key role in the team as solution provider for simplifying the existing system while working for clients saved cost and time per annum by automating administrative system wrote technical documents and user 's manuals technical summary languages skills python java skills angular js polymer 1.0 jquery library and tool tensorflow deep learning machine learning word2vec artificial intelligence pycharm jupiter notebook frameworks django web2py flask jpa google app engine platform google apps integration databases skills mongodb google big table(nosql database mysql 5.0 sqlite postgresql google cloud sql redis operating systems linux(ubuntu window applications google apps webex database zoho ms office google apps migration from lotus notes working with active directory syncing with the google apps elastic search spring mvc gantter product b1 b2 us visa till 2025 education mongo dadabari ajmer rajasthan june 2015 to august 2016 bca degree affiliated college_name kota rajasthan https://www.indeed.com/r/ramkrishan-bhatt/da07dc6d058dfc64?isid=rex-download&ikw=download-top&co=in https://www.indeed.com/r/ramkrishan-bhatt/da07dc6d058dfc64?isid=rex-download&ikw=download-top&co=in

In [119...
```python
spacy.__version__
```

Out[119... '3.1.4'

In [ ]: