Name_entity_classifier

May 10, 2022

0.1 my approach is to build a custom NER model for store number

0.2 Custom NER using Spacy

- **Features Extraction:** From every transaction_descriptor I extract the first and last index the store number appear in. So, the input to the model is (start_index,end_index,label). in our case the label is ”StoreNumber”
- **Data preperation:** As I use spacy model(Rule based model) to be trainned in our data. The model accept only ***.spacy*** files.So, I convert to .spacy

```
[ ]: !pip install spacy==3.0.6
```

```
[4]: from google.colab import drive
     drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[6]: import pandas as pd
     import re
     import spacy
     from spacy.tokens import DocBin
     from tqdm import tqdm
     from spacy.util import filter_spans
     import numpy as np

     df=pd.read_csv("/content/drive/MyDrive/Summer Internship - Homework Exercise.
      ↪csv")
     df
```

```
[6]:              transaction_descriptor store_number dataset
     0    DOLRTREE 2257 00022574 ROSWELL         2257   train
     1                    AUTOZONE #3547         3547   train
     2              TGI FRIDAYS 1485 0000         1485   train
     3             BUFFALO WILD WINGS 003            3   train
     4                    J. CREW #568 0          568   train
     ..                                …            …     …
     295                  MCDONALD'S F2151        F2151    test
```

```
296       NST BEST BUY #1403 332411         1403    test
297             CVS/PHARMACY #06689         6689    test
298           BANANA REPUBLIC #8109         8109    test
299              BOSTON MARKET 0443          443    test

[300 rows x 3 columns]
```

0.3 Data Splliting

[7]:
```
df_train=df[df["dataset"]=="train"]
df_valid=df[df["dataset"]=="validation"]
df_test=df[df["dataset"]=="test"]

df_train
```

[7]:
```
            transaction_descriptor store_number dataset
0   DOLRTREE 2257 00022574 ROSWELL         2257   train
1                   AUTOZONE #3547         3547   train
2              TGI FRIDAYS 1485 0000         1485   train
3             BUFFALO WILD WINGS 003            3   train
4                   J. CREW #568 0          568   train
..                               …            …     …
95                  DUNKIN #355514       355514   train
96             THE HOME DEPOT #6828         6828   train
97                 RUE21 #1129 BLUE         1129   train
98               WM SUPERCENTER #34           34   train
99          RACETRAC485   00004853          485   train

[100 rows x 3 columns]
```

[28]:
```
print(df_train["store_number"].apply(lambda x : len(x)).max())
print(df_valid["store_number"].apply(lambda x : len(x)).max())
print(df_test["store_number"].apply(lambda x : len(x)).max())
```

```
9
9
9
```

[ ]:
```
#downlad it  for utils GPU during training
!pip install spacy[transformers]
```

0.3.1 Feature Extraction

[19]:
```
def extract_text_spans(df):
  L = []
  for i in range(len(df)):
    S = df['transaction_descriptor'].iloc[i]
    w = df['store_number'].iloc[i]
    match = re.search( w, S)
    start = match.start()
    end = match.end()
    l = [S ,w ,start , end]
    L.append(l)


  data = pd.DataFrame( L , columns=['transaction_descriptor', 'store_number' ,␣
↪'start', 'end'])
  return data
```

[35]:
```
train=extract_text_spans(df_train)
valid=extract_text_spans(df_valid)
```

[21]:
```
train
```

[21]:
```
            transaction_descriptor store_number  start  end
0   DOLRTREE 2257 00022574 ROSWELL         2257      9   13
1                  AUTOZONE #3547         3547     10   14
2              TGI FRIDAYS 1485 0000         1485     12   16
3             BUFFALO WILD WINGS 003            3     21   22
4                   J. CREW #568 0          568      9   12
..                             …            …    …  …
95                  DUNKIN #355514       355514      8   14
96              THE HOME DEPOT #6828         6828     16   20
97                 RUE21 #1129 BLUE         1129      7   11
98               WM SUPERCENTER #34           34     16   18
99            RACETRAC485    00004853          485      8   11

[100 rows x 4 columns]
```

[22]:
```
nlp = spacy.blank("en")
doc_bin = DocBin()
```

0.3.2 Data preparation

[23]:
```
def create_spacy_file(df,file_name):
  skipped=0
  for i  in tqdm(range(len(df))):
      text = df["transaction_descriptor"].iloc[i]
      doc = nlp.make_doc(text)
      ents = []
      span = doc.char_span(df['start'].iloc[i],df['end'].iloc[i], label=␣
↪'StoreNumber', alignment_mode="contract")
      if span is None:
            skipped+=1
      else:
            ents.append(span)
      filtered_ents = filter_spans(ents)
      doc.ents = filtered_ents
      doc_bin.add(doc)

  doc_bin.to_disk(f"{file_name}.spacy")
  print(f"\nskipped {file_name} entities={skipped}\n")
```

[24]:
```
create_spacy_file(train,"train")
create_spacy_file(valid,"validation")
```

```
100%|      | 100/100 [00:00<00:00, 1176.38it/s]


skipped train entities=34


100%|      | 100/100 [00:00<00:00, 1100.89it/s]


skipped validation entities=39
```

0.3.3 Training the Model

[25]:
```
!python -m spacy init config --lang en --pipeline ner config.cfg --force
```

```
 Generated config template specific for your use case
- Language: en
- Pipeline: ner
- Optimize for: efficiency
- Hardware: CPU
- Transformer: None
```

```
  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
```

[26]:
```
!python -m spacy train config.cfg --output ./ --paths.train ./train.spacy␣
  ↪--paths.dev ./train.spacy
```

```
  Using CPU
  To switch to GPU 0, use the option: --gpu-id 0


=========================== Initializing pipeline

===========================
[2022-05-10 03:46:07,361] [INFO] Set up nlp object from config
[2022-05-10 03:46:07,373] [INFO] Pipeline: ['tok2vec', 'ner']
[2022-05-10 03:46:07,378] [INFO] Created vocabulary
[2022-05-10 03:46:07,378] [INFO] Finished initializing nlp object
[2022-05-10 03:46:07,585] [INFO] Initialized pipeline components: ['tok2vec',
'ner']
  Initialized pipeline


============================= Training pipeline

=============================
  Pipeline: ['tok2vec', 'ner']
  Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     64.00    0.00    0.00    0.00    0.00
 50     200          3.31    727.12  100.00  100.00  100.00    1.00
113     400          0.00      0.00  100.00  100.00  100.00    1.00
180     600          0.00      0.00  100.00  100.00  100.00    1.00
277     800          0.00      0.00  100.00  100.00  100.00    1.00
377    1000          0.00      0.00  100.00  100.00  100.00    1.00
477    1200          0.00      0.00  100.00  100.00  100.00    1.00
670    1400          0.00      0.00  100.00  100.00  100.00    1.00
870    1600         50.08     11.66  100.00  100.00  100.00    1.00
1070     1800          19.81       3.98  100.00  100.00  100.00    1.00
  Saved pipeline to output directory
model-last
```

[27]:
```
!python -m spacy train config.cfg --output ./ --paths.train ./validation.spacy␣
  ↪--paths.dev ./train.spacy
```

```
  Using CPU
```

```
 To switch to GPU 0, use the option: --gpu-id 0

=========================== Initializing pipeline
===========================
[2022-05-10 03:50:46,379] [INFO] Set up nlp object from config
[2022-05-10 03:50:46,391] [INFO] Pipeline: ['tok2vec', 'ner']
[2022-05-10 03:50:46,396] [INFO] Created vocabulary
[2022-05-10 03:50:46,396] [INFO] Finished initializing nlp object
[2022-05-10 03:50:46,691] [INFO] Initialized pipeline components: ['tok2vec',
'ner']
 Initialized pipeline

============================= Training pipeline
=============================
 Pipeline: ['tok2vec', 'ner']
 Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     62.83    0.00    0.00    0.00    0.00
 25     200          9.21    726.38  100.00  100.00  100.00    1.00
 55     400          0.00      0.00  100.00  100.00  100.00    1.00
 92     600          0.00      0.00  100.00  100.00  100.00    1.00
137     800          0.00      0.00  100.00  100.00  100.00    1.00
189    1000          0.00      0.00  100.00  100.00  100.00    1.00
256    1200         88.97     24.24  100.00  100.00  100.00    1.00
332    1400         41.58     11.53  100.00  100.00  100.00    1.00
432    1600          0.00      0.00  100.00  100.00  100.00    1.00
532    1800          0.00      0.00  100.00  100.00  100.00    1.00
 Saved pipeline to output directory
model-last
```

0.3.4 Testing model and visulaize the results

[37]:
```python
nlp_ner = spacy.load("model-best")
preds=[]
matched=0
for i in range(len(df_test)):
  doc = nlp_ner(df_test["transaction_descriptor"].iloc[i])
  colors = {"StoreNumber": "#7DF6D9"}
  options = {"colors": colors}
  if  len(doc.ents) >= 1 and doc.ents[0].label_ == "StoreNumber":
        preds.append(str(doc.ents[0]))
        doc.ents[0].label_ == "StoreNumber"
        spacy.displacy.render(doc, style="ent", options= options, jupyter=True)
```

```
            matched+=1
    else:
        preds.append('Not matched')
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

[30]:

```
print(f"extracted True sample = {matched}\nNot Extracted sample =␣
 ↪{100-matched}")
```

```
extracted True sample = 72
Not Extracted sample = 28
```

0.3.5 the model extract 72 store number out of 100 entries

```
[38]: np.sum(np.equal(df_test['store_number'], preds))
```

```
[38]: 71
```

0.3.6 71 of 72 extracted number are correctly identified as Store number

```
[33]: test_preds = pd.DataFrame({'test' : df_test['store_number'] , 'preds': preds})
      test_preds
```

```
[33]:         test        preds
      200      242          242
      201  9442088  Not matched
      202     1419         1419
      203     1019         1019
      204       38           38
      ..       …           …
      295    F2151        F2151
      296     1403         1403
      297     6689  Not matched
      298     8109         8109
      299      443  Not matched

      [100 rows x 2 columns]
```

0.4 model accuracy

```
[34]: import numpy as np
      acc = np.sum(np.equal(df_test['store_number'], preds)) /␣
       ↪len(df_test['store_number'])
      acc
```

```
[34]: 0.71
```

```
[ ]:
```

```
[ ]: !wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
     from colab_pdf import colab_pdf
     colab_pdf('Name_entity_classifier.ipynb')
```

```
File 'colab_pdf.py' already there; not retrieving.


WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
```

```
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
```