Natural Language Processing

Project#3 Person Name Recognition

GROUP #19

LI JIALIN D-B9-2592-2

ZHANG HUAKANG D-B9-2760-6

Introduction

In this project, we build a maximum entropy model (MEM) for identifying person names ('Named Entity', NER) in newswire texts and it achieves a very high performance by a set of features of the input words. We notice that whether a word is a name not only depends on itself, but also its neighbors. Based on this observation, we choose the part of speech of the word and its neighbors as the main features in our model. We also built a front-end website for this project and make the source code public on GitHub.

https://nlpproject.boxz.dev

https://github.com/BoxMars/NLP_Project/tree/master/Project3

Methods

- For our approach, we directly use the part of speech of the input word and its neighbors in a sentence,
- $w_{n-2}^{n+2} = \{w_{n-2}, w_{n-1}, w_n, w_{n+1}, w_{n+2}\}$.
- The basic feature list contains:
  - w_n
  - $lable(w_{n-1})$
  - $isUpper(w_n[0])$

Methods

- The features we add:
  - $isAlpha(w_n)$
  - $isPeriod(w_n)$
  - w_{n-2}
  - $pos(w_{n-2})$
  - w_{n-1}
  - $pos(w_{n-1})$
  - w_{n+1}
  - $pos(w_{n+1})$
  - w_{n+2}
  - $pos(w_{n+2})$
  - where $pos(\cdot)$ is the function that get the part of speech of the word.

Methods

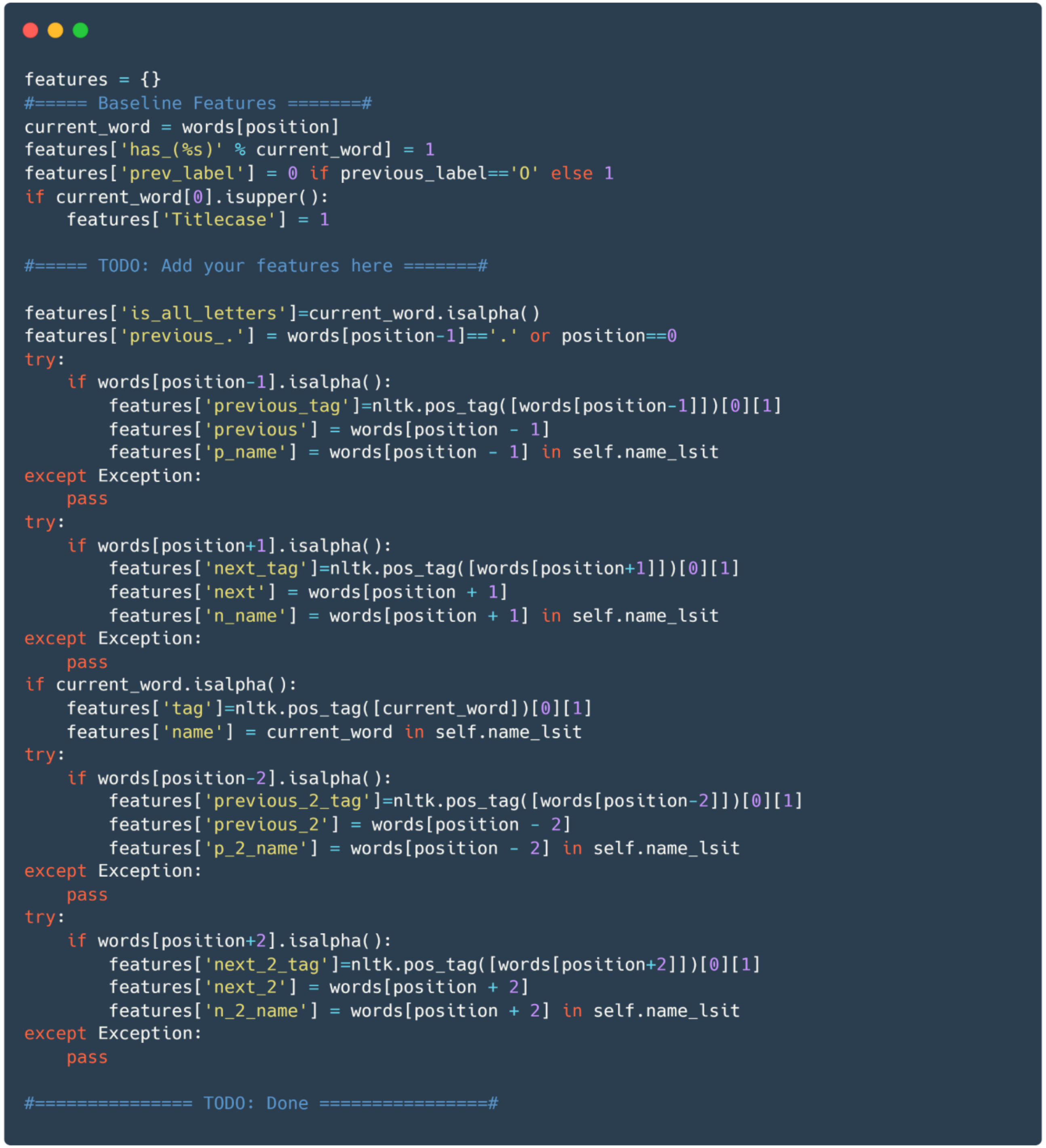
- But only the part of speech is not enough for NER since the name word can be replaced with any nouns.
- For example,
- *President Biden today agrees to send weapons to Ukraine*
- *US Congress today agrees to send weapons to Ukraine*
- have same sentence structure.
- If we only use the part of speech of the target word and its neighbors, this model will become noun recognition instead of the person’s name recognition.

Methods

- Thus, we consider using `nltk.corpus.name` to check if the word is a name word to enhance our model. The name feature list contains:
- $isInNameCorpus(w_n)$
- $isInNameCorpus(w_{n-2})$
- $isInNameCorpus(w_{n-1})$
- $isInNameCorpus(w_{n+1})$
- $isInNameCorpus(w_{n+1})$

Implementation

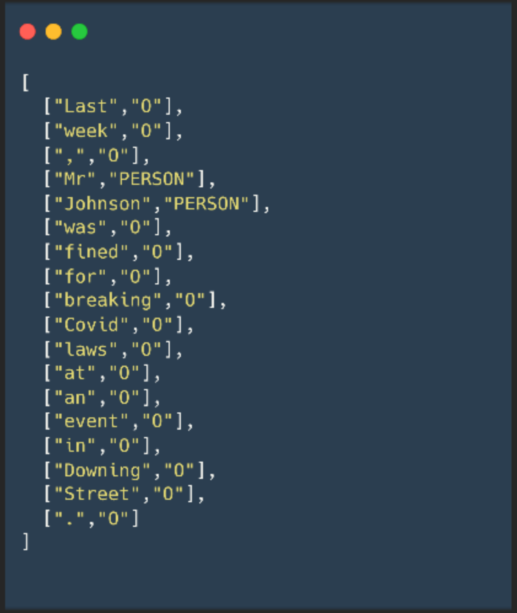
NER Model



```
features = {}
#====== Baseline Features =======#
current_word = words[position]
features['has_(%s)' % current_word] = 1
features['prev_label'] = 0 if previous_label=='O' else 1
if current_word[0].isupper():
    features['Titlecase'] = 1

#====== TODO: Add your features here =======#

features['is_all_letters']=current_word.isalpha()
features['previous_.'] = words[position-1]=='.' or position==0
try:
    if words[position-1].isalpha():
        features['previous_tag']=nltk.pos_tag([words[position-1]])[0][1]
        features['previous'] = words[position - 1]
        features['p_name'] = words[position - 1] in self.name_lsit
except Exception:
    pass
try:
    if words[position+1].isalpha():
        features['next_tag']=nltk.pos_tag([words[position+1]])[0][1]
        features['next'] = words[position + 1]
        features['n_name'] = words[position + 1] in self.name_lsit
except Exception:
    pass
if current_word.isalpha():
    features['tag']=nltk.pos_tag([current_word])[0][1]
    features['name'] = current_word in self.name_lsit
try:
    if words[position-2].isalpha():
        features['previous_2_tag']=nltk.pos_tag([words[position-2]])[0][1]
        features['previous_2'] = words[position - 2]
        features['p_2_name'] = words[position - 2] in self.name_lsit
except Exception:
    pass
try:
    if words[position+2].isalpha():
        features['next_2_tag']=nltk.pos_tag([words[position+2]])[0][1]
        features['next_2'] = words[position + 2]
        features['n_2_name'] = words[position + 2] in self.name_lsit
except Exception:
    pass

#=============== TODO: Done ================#
```

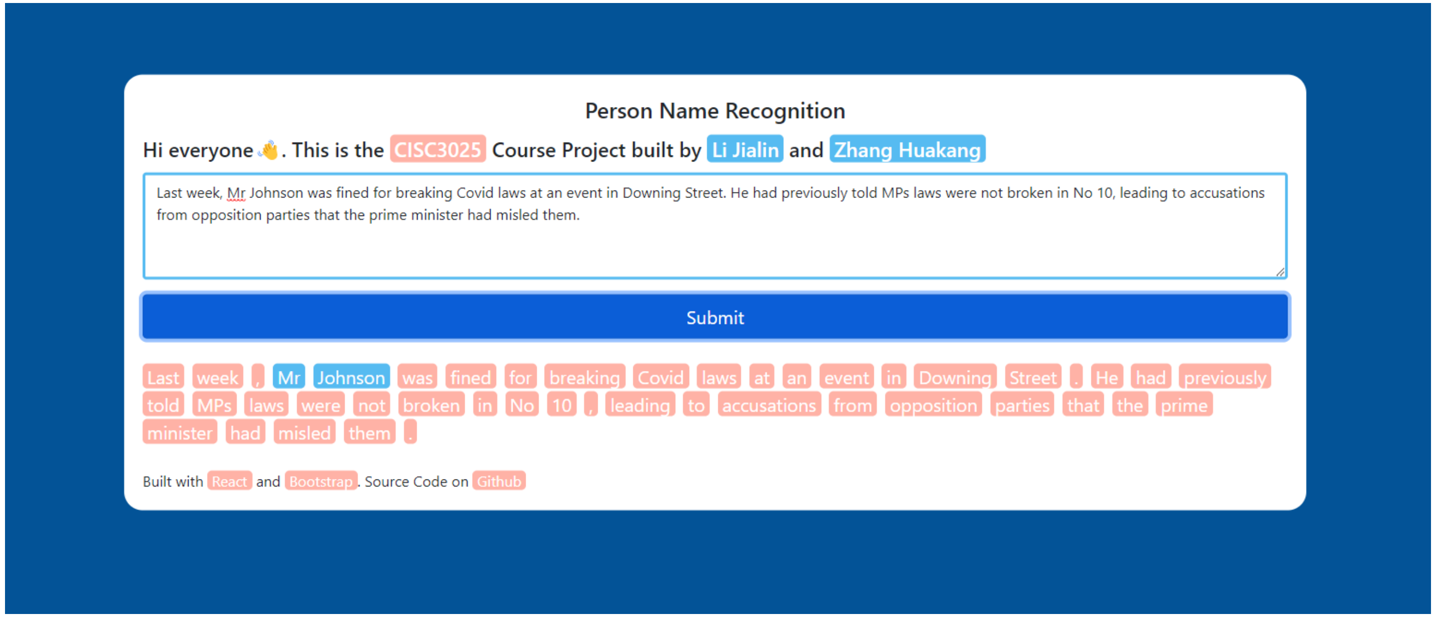
Implementation

Web Server



```
[
  ["Last","O"],
  ["week","O"],
  [",","O"],
  ["Mr","PERSON"],
  ["Johnson","PERSON"],
  ["was","O"],
  ["fined","O"],
  ["for","O"],
  ["breaking","O"],
  ["Covid","O"],
  ["laws","O"],
  ["at","O"],
  ["an","O"],
  ["event","O"],
  ["in","O"],
  ["Downing","O"],
  ["Street","O"],
  [".","O"]
]
```

We use `flask` package to develop the `API` server and built a front-end website with `React` and `Bootstrap`. You can access `https://nlpproject.boxz.dev` to experience our project or access `https://nlpproject.boxz.dev/api/?text=<sentence>` to experience the back-end API.



Person Name Recognition

Hi everyone 👋. This is the CISC3025 Course Project built by Li Jialin and Zhang Huakang

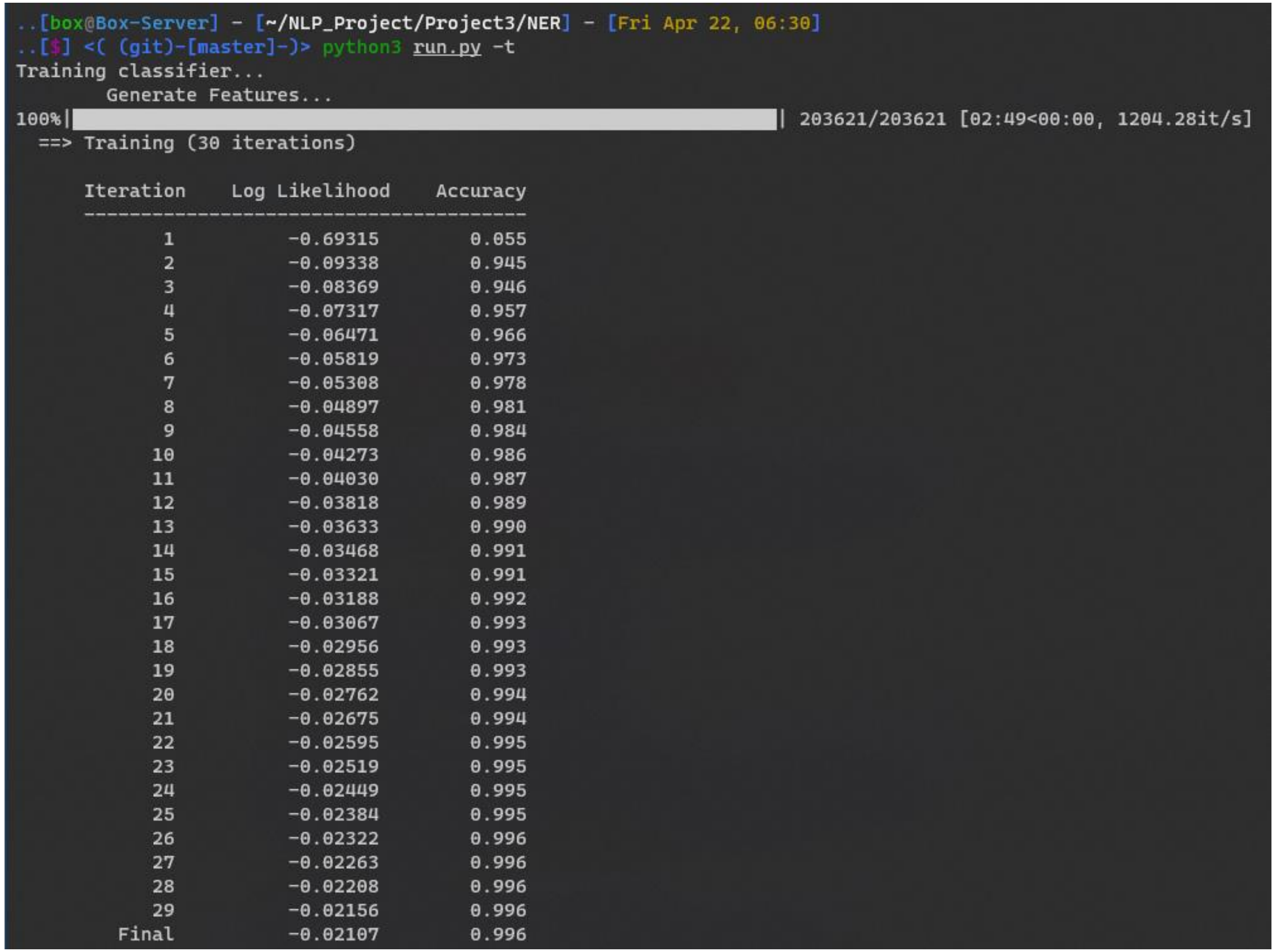
Last week, Mr Johnson was fined for breaking Covid laws at an event in Downing Street. He had previously told MPs laws were not broken in No 10, leading to accusations from opposition parties that the prime minister had misled them.

Last week , Mr Johnson was fined for breaking Covid laws at an event in Downing Street . He had previously told MPs laws were not broken in No 10 , leading to accusations from opposition parties that the prime minister had misled them .

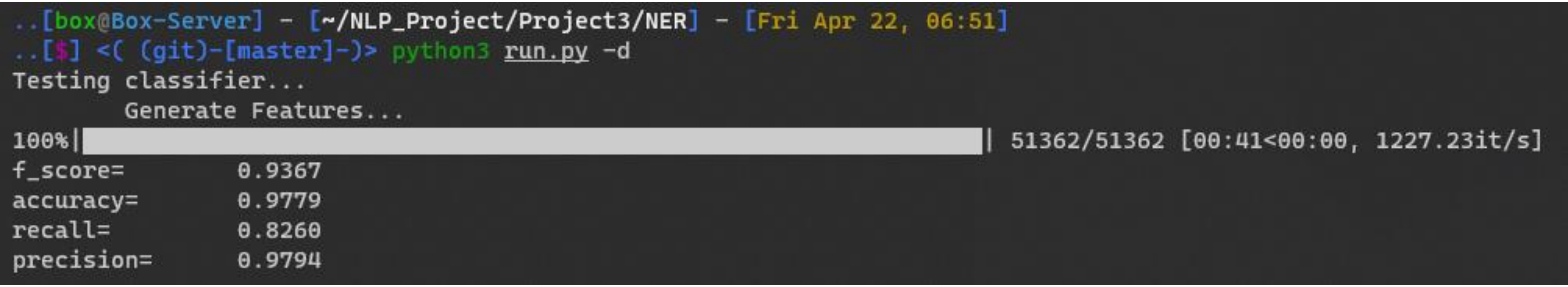
Built with React and Bootstrap. Source Code on Github

Evaluations

Training



```
..[box@Box-Server] - [~/NLP_Project/Project3/NER] - [Fri Apr 22, 06:30]
..[$] <( (git)-[master]-)> python3 run.py -t
Training classifier...
        Generate Features...
100%|████████████████████████████████████████████████████████████| 203621/203621 [02:49<00:00, 1204.28it/s]
  ==> Training (30 iterations)

      Iteration    Log Likelihood    Accuracy
      ---------------------------------------
             1          -0.69315        0.055
             2          -0.09338        0.945
             3          -0.08369        0.946
             4          -0.07317        0.957
             5          -0.06471        0.966
             6          -0.05819        0.973
             7          -0.05308        0.978
             8          -0.04897        0.981
             9          -0.04558        0.984
            10          -0.04273        0.986
            11          -0.04030        0.987
            12          -0.03818        0.989
            13          -0.03633        0.990
            14          -0.03468        0.991
            15          -0.03321        0.991
            16          -0.03188        0.992
            17          -0.03067        0.993
            18          -0.02956        0.993
            19          -0.02855        0.993
            20          -0.02762        0.994
            21          -0.02675        0.994
            22          -0.02595        0.995
            23          -0.02519        0.995
            24          -0.02449        0.995
            25          -0.02384        0.995
            26          -0.02322        0.996
            27          -0.02263        0.996
            28          -0.02208        0.996
            29          -0.02156        0.996
         Final          -0.02107        0.996
```



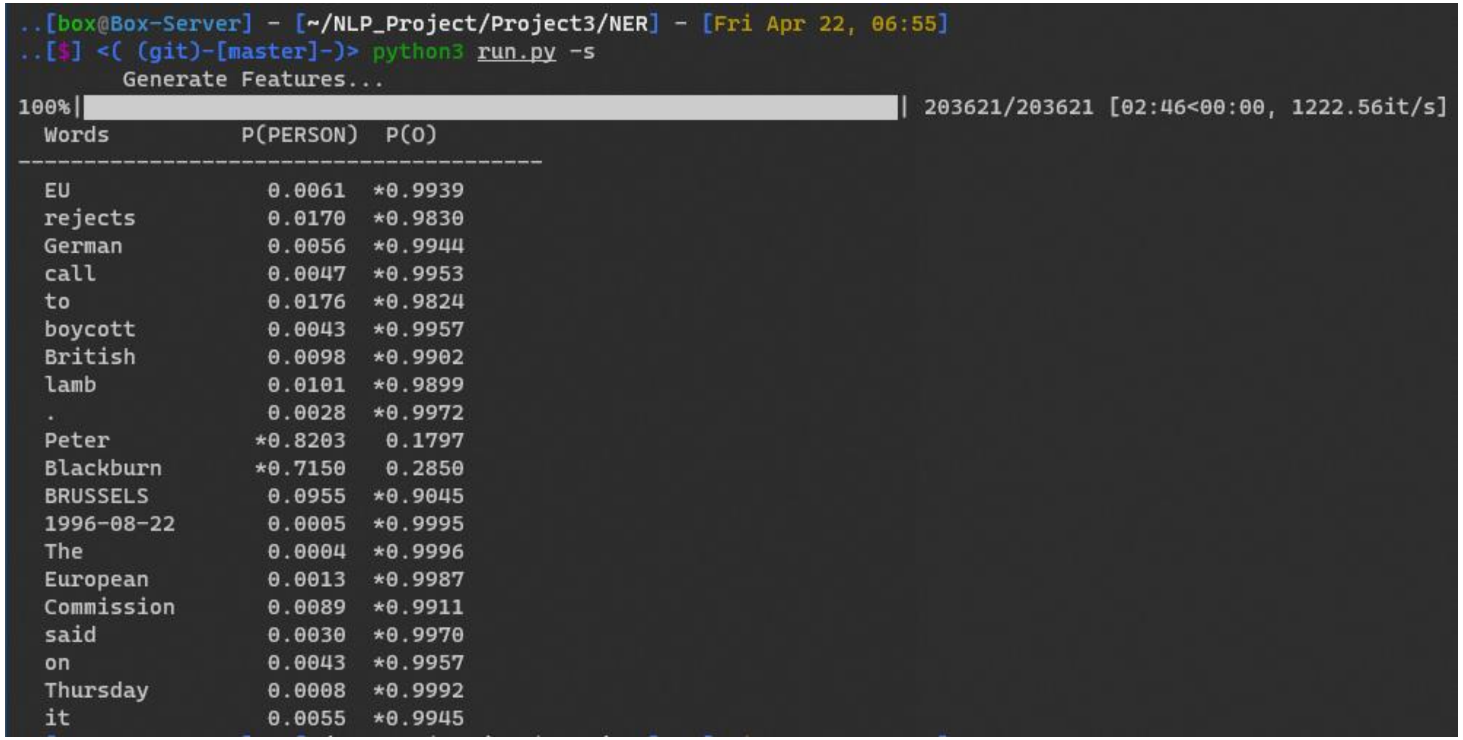
```
..[box@Box-Server] - [~/NLP_Project/Project3/NER] - [Fri Apr 22, 06:51]
..[$] <( (git)-[master]-)> python3 run.py -d
Testing classifier...
        Generate Features...
100%|████████████████████████████████████████████████████████████████| 51362/51362 [00:41<00:00, 1227.23it/s]
f_score=        0.9367
accuracy=       0.9779
recall=         0.8260
precision=      0.9794
```

Evaluations

Testing

Evaluations

Output



```
..[box@Box-Server] - [~/NLP_Project/Project3/NER] - [Fri Apr 22, 06:55]
..[$] <( (git)-[master]-)> python3 run.py -s
        Generate Features...
100%|██████████████████████████████████████████████████████████████| 203621/203621 [02:46<00:00, 1222.56it/s]
  Words          P(PERSON)  P(O)
----------------------------------------
  EU               0.0061  *0.9939
  rejects          0.0170  *0.9830
  German           0.0056  *0.9944
  call             0.0047  *0.9953
  to               0.0176  *0.9824
  boycott          0.0043  *0.9957
  British          0.0098  *0.9902
  lamb             0.0101  *0.9899
  .                0.0028  *0.9972
  Peter           *0.8203   0.1797
  Blackburn       *0.7150   0.2850
  BRUSSELS         0.0955  *0.9045
  1996-08-22       0.0005  *0.9995
  The              0.0004  *0.9996
  European         0.0013  *0.9987
  Commission       0.0089  *0.9911
  said             0.0030  *0.9970
  on               0.0043  *0.9957
  Thursday         0.0008  *0.9992
  it               0.0055  *0.9945
```

Evaluations

Discussions

We notice that when we try adding a lot of features in this model, the generating process will take a long time. It is easy to know that the generating f each is independent, actuarily it is possible that use multithreading or metaprograms to accelerate this process. But during to the Global Interpreted Lock (GIL) in python, multithreading may not work limited by the clock speed of CPU.

Conclusion

In this project, we build a maximum entropy model (MEM) for identifying person names ('Named Entity', NER) in newswire texts and it achieves a very high performance by a set of features of the input words. We also set up a front-end website for this model for visualization.