Hidden Markov Model and Viterbi Algorithm

Question 1

replace infrequent words with _RARE_

- implemented in **src/replace_with_rare.py**
- run:

```
$ python replace_with_rare.py
$ python count_freqs.py  ../output/ner_train_with_rare.dat >  ../output/ner_with_rare.counts
```

- output file path:
  - updated data file with *RARE*: **output/ner_train_with_rare.dat**
  - updated count file with *RARE*: **output/ner_with_rare.counts**

compute emission parameters

- implemented in **src/baseline.py** as `compute_emission(word, ner)`

Question 2

baseline NER tagger

- implemented in **src/baseline.py**
- run:

```
$python baseline.py
```

- output file path: **output/dev_tagged.predict**
- evaluation:
  - run: `$ python eval_ne_tagger.py ../input/ner_dev.key ../output/dev_tagged.predict`
  - result:

```
Found 14043 NEs. Expected 5931 NEs; Correct: 3117.

         precision      recall          F1-Score
Total:   0.221961       0.525544        0.312106
PER:     0.435451       0.231230        0.302061
ORG:     0.475936       0.399103        0.434146
LOC:     0.147750       0.870229        0.252612
MISC:    0.491689       0.610206        0.544574
```

Question 3

compute the log probability of trigrams

- Implemented in **src/trigram.py**
- Test:
  - I made up ten trigrams and stored them in **input/trigrams_test.dat**
  - boundary cases below were included
    - q (y1|*, *)
    - q (y2|*, y1)
    - q (STOP|y(n-1), yn)
  - run: `$ python trigram.py`
  - output path: **output/trigrams_test.dat**

Question 4

implement viterbi algorithm

- implemented in **src/viterbi.py**
- run:

```
$ python viterbi.py
```

- output path: **output/sen_tagged.predict**

evaluation

- `$ python eval_ne_tagger.py ../input/ner_dev.key ../output/sen_tagged.predict`
- result:

```
Found 4704 NEs. Expected 5931 NEs; Correct: 3643.

         precision      recall          F1-Score
Total:   0.774447       0.614230        0.685096
PER:     0.759749       0.593580        0.666463
ORG:     0.611855       0.478326        0.536913
LOC:     0.876458       0.696292        0.776056
MISC:    0.830065       0.689468        0.753262
```