Ricardo Rendon

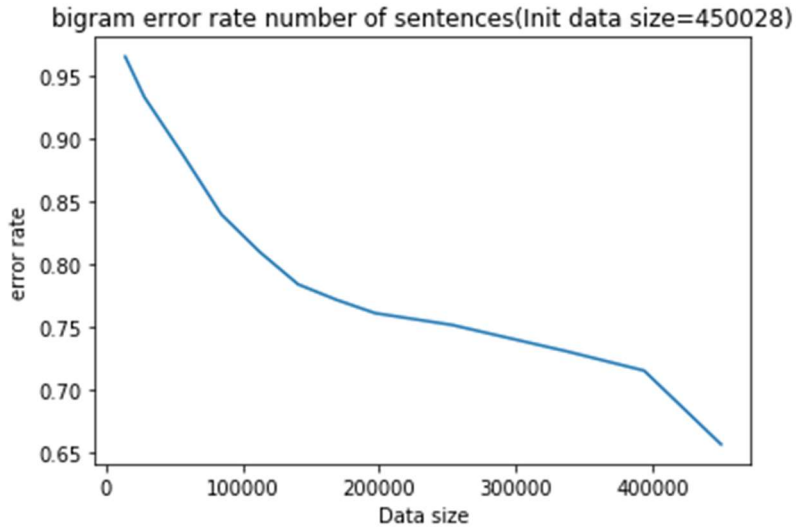
Task 1:

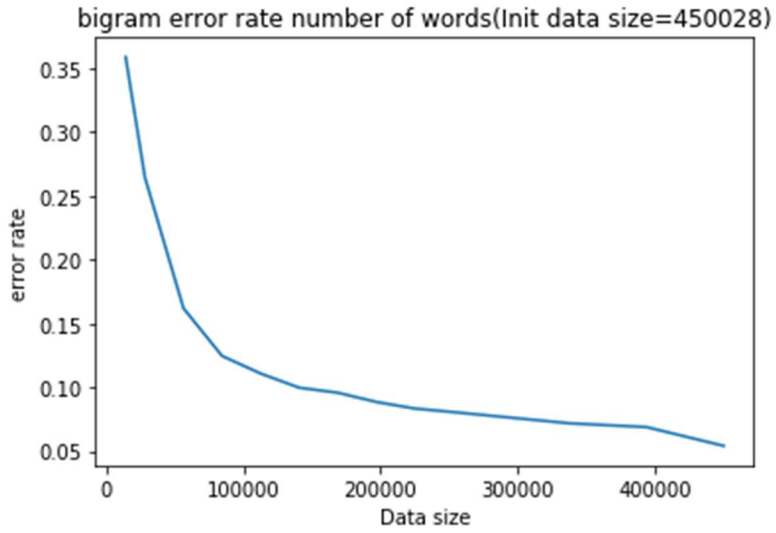
By addding a coondition to my code:

#counternew=counternew+1

#if counternew > (450028/x)*x:continue

I could use part of the data to generate the tags.





Task2:

- In code: “viterbi.bigram.py”
- Tag generated= ‘my.out.bigram’

Ricardo Rendon

Task3:

I did a trigram with backoff to the bigram output

- Train_hmm.trigram.py
- Viterbi.trigram.py
    - Note: needs to run Viterbi bigram code since it uses back off model and uses the generated bigram tags. In code opens 'my.out.bigram' tags generated files from the top task
- Tags generated: my.out.trigram
- To generates tags for 23txt use

After training with text21 and tag 21 running the code with 22txt and checking with the tags I got an error of:

```
C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>perl tag_acc.pl ptb.22.tgs my.out.trigram
error rate by word:      0.0500535932397737 (2008 errors out of 40117)
error rate by sentence:  0.625294117647059 (1063 errors out of 1700)
```

- error rate by word: 0.0500535932397737 (2008 errors out of 40117)
- error rate by sentence: 0.625294117647059 (1063 errors out of 1700)

which is better than the prof error rates and it is above the required threshold(error rate by word: 0.0518 error rate by sentence: 0.641)

For the 23 text here is the code to create the tags.(tags already included)

```
C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.bigram.py my.hmm.bigram  ptb.23.txt > my.out.bigram23tg

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.trigram.py my.hmm.trigram  ptb.23.txt > my.out.trigram23tg

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.trigram.23.txt.py my.hmm.trigram  ptb.23.txt > my.out.trigram23tg

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>
```

Task4:

For this task, we wanted to see if our method of determining tags are as useful In English as for other languages.

We utilize the given bigram model to generate tags in jv and in btb:

Jv:

```
C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>perl tag_acc.pl jv.test.tgs my.out.bigramjv
error rate by word:      0.0628611451584661 (359 errors out of 5711)
error rate by sentence:  0.136812411847673 (97 errors out of 709)
```

Btb:

```
C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>perl train_hmm.pl btb.train.tgs btb.train.txt > my.hmm.bigrambtb

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>perl viterbi.pl my.hmm.bigrambtb < btb.test.txt > my.out.bigrambtb

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>perl tag_acc.pl btb.test.tgs my.outbigrambtb
can't open file my.outbigrambtb at tag_acc.pl line 20.

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>perl tag_acc.pl btb.test.tgs my.out.bigrambtb
error rate by word:      0.115942028985507 (688 errors out of 5934)
error rate by sentence:  0.751256281407035 (299 errors out of 398)

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>
```

Ricardo Rendon

With my model for jv and btb:

```
C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>train_hmm.trigram.py jv.train.tgs jv.train.txt > my.hmm.trigramjv

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.trigram.jv.py my.hmm.trigramjv < jv.test.txt > my.out.trigramjv
Traceback (most recent call last):
  File "C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2\viterbi.trigram.jv.py", line 51, in <module>
    if trans_dic[tuple([w,u,v])]!=-9999 and emit_dic[tuple([v,word])]!=-9999 and pi[tuple([count-1,w,u])]!=-9999:
KeyboardInterrupt

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.trigram.jv.py my.hmm.trigramjv < jv.test.txt > my.out.trigramjv
Traceback (most recent call last):
  File "C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2\viterbi.trigram.jv.py", line 50, in <module>
    for v in Pos_tags:
KeyboardInterrupt

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.trigram.jv.py my.hmm.trigramjv < jv.test.txt > my.out.trigramjv

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>perl tag_acc.pl jv.test.tgs my.out.bigramjv
error rate by word:      0.0628611451584661 (359 errors out of 5711)
error rate by sentence:  0.136812411847673 (97 errors out of 709)
```

```
C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>train_hmm.trigram.btb.py btb.train.tgs btb.train.txt > my.hmm.trigrambtb

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.trigram.btb.py my.hmm.trigrambtb < btb.test.txt > my.out.trigrambtb
Traceback (most recent call last):
  File "C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2\viterbi.trigram.btb.py", line 51, in <module>
    if trans_dic[tuple([w,u,v])]!=-9999 and emit_dic[tuple([v,word])]!=-9999 and pi[tuple([count-1,w,u])]!=-9999:
KeyboardInterrupt

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.trigram.btb.py my.hmm.trigrambtb < btb.test.txt > my.out.trigrambtb
Traceback (most recent call last):
  File "C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2\viterbi.trigram.btb.py", line 51, in <module>
    if trans_dic[tuple([w,u,v])]!=-9999 and emit_dic[tuple([v,word])]!=-9999 and pi[tuple([count-1,w,u])]!=-9999:
KeyboardInterrupt

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.trigram.btb.py my.hmm.trigrambtb < btb.test.txt > my.out.trigrambtb
Traceback (most recent call last):
  File "C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2\viterbi.trigram.btb.py", line 51, in <module>
    if trans_dic[tuple([w,u,v])]!=-9999 and emit_dic[tuple([v,word])]!=-9999 and pi[tuple([count-1,w,u])]!=-9999:
KeyboardInterrupt

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>viterbi.trigram.btb.py my.hmm.trigrambtb < btb.test.txt > my.out.trigrambtb

C:\Users\rire9\OneDrive\Escritorio\AI\hw2\ECS 189G HW 2>perl tag_acc.pl btb.test.tgs my.out.trigrambtb
error rate by word:      0.115942028985507 (688 errors out of 5934)
error rate by sentence:  0.751256281407035 (299 errors out of 398)
```

Using error rate by word we can see that the model perform the best in Japanese and English. One reason could be that words in these 2 languages rely more on the words that are close by(since we using bigram or trigram). In the other hand Bulgarian the words that are close by might not have such an impact and another type of structure could be in place. I do not know what kind of structure the language has since I don't know the language but it seems that close by words(tags) might behave differently.