Natural Language Processing, Homework 4

Jesse Weller, Sheekha Jariwala, Prachi Rahurkar

Monday November 18, 2019

1 Complete Data, Incomplete Model (10 pts)

If we're given the following manually aligned pairs (English "boat" and "test"):

```
B OW T          T EH S T
B O O T O       T E S U T O
1 2 2 3 3       1 2 3 3 4 4
```

What is the maximum likelihood estimate for $p(\texttt{T} \mid \texttt{T})$ and $p(\texttt{T O} \mid \texttt{T})$?

The maximum likelihood estimate for $p(\texttt{T} \mid \texttt{T})$ is = count(T:T) / count(T) .i.e. 1/3 which is 0.333 and The maximum likelihood estimate for $p(\texttt{T O} \mid \texttt{T})$ is = count(T:TO) / count(T) .i.e. 2/3 which is 0.666.

2 Complete Model, Incomplete Data (10 pts)

Given the following conditional probabilities,

```
p(B | B) = 1
p(O O | OW) = 0.8   p(O | OW) = 0.2
p(T | T) = 0.3      p(T O | T) = 0.5   p(O | T) = 0.1   p(O T O | T) = 0.1
```

Enumerate (by hand) all legal alignments for the "boat" example above, and compute the probability and normalized probability of each alignment. Which one is the Viterbi alignment?

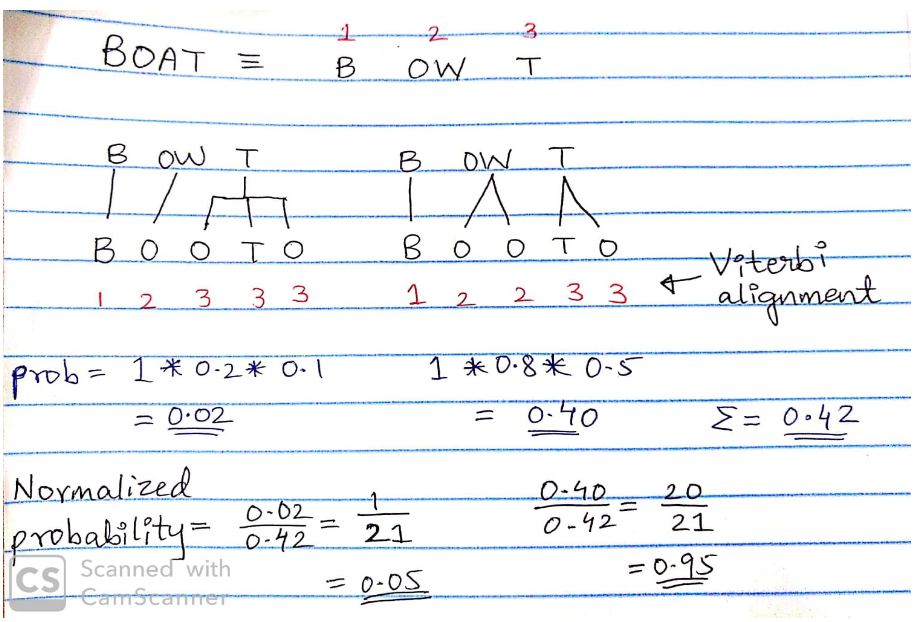

Figure 1: viterbi alignment as shown

3 EM, Implementation I: Enumerating All Alignments (70 pts)

Now implement the EM algorithm:

1. initialize the conditional probabilities to uniform
2. repeat:

3. E-step:

4. for each English-Katakana pair:

5. enumerate all legal alignments for that E-K pair, along with their respective probabilities

6. renormalize the probabilities to get a distribution over these alignments

7. collect the fractional counts for this E-K pair

8. M-step: count-and-divide based on the collected fractional counts from all pairs to get new prob. table

9. until reached maximum iteration or converged

You should proceed with the following steps:

1. (15 pts) to start with, you should work on the simplest scenario ("wine" example): `W AY N / W A I N`.

   Print the following information in each iteration: 1) the corpus probability, 2) the new prob. table (>= 0.01). 3) the number of nonzero entries in the prob. table (>= 0.01). **Note:** the corpus probability should increase in each iteration (we have the proof), and the number of nonzero entries should decrease (as the model gets peakier).

   For example, you should run `echo -e "W AY N\nW A I N\n" | ./em.py 5` and get the following (note that the data should be read in from standard input and 5 is the number of iterations):

```
iteration 0  ----- corpus prob=  0.00411522633745
AY|->   A: 0.33  A I: 0.33  I: 0.33
W|->    W: 0.67  W A: 0.33
N|->    N: 0.67  I N: 0.33
nonzeros = 7

iteration 1  ----- corpus prob=  0.296296296296
AY|->   A I: 0.50  A: 0.25  I: 0.25
W|->    W: 0.75  W A: 0.25
N|->    N: 0.75  I N: 0.25
nonzeros = 7
...
iteration 4  ----- corpus prob=  0.912659654395
AY|->   A I: 1.00
W|->    W: 1.00
N|->    N: 1.00
nonzeros = 3
```

   For your debugging convenience you can also print all possible alignments and their unnormalized and normalized probabilities in each iteration. Include the result of 5 iterations in your report.

   The result of first 5 iterations:

   command: cat wayn.txt — python em.py 5

```
iteration 0  ---- corpus prob=  0.004115226337448556
W|->  W A: 0.333  W: 0.667
AY|->  I: 0.333  A I: 0.333  A: 0.333
N|->  N: 0.667  I N: 0.333
nonzeros: 7


iteration 1  ---- corpus prob=  0.29629629629629634
W|->  W A: 0.250  W: 0.750
AY|->  I: 0.250  A I: 0.500  A: 0.250
N|->  N: 0.750  I N: 0.250
nonzeros: 7


iteration 2  ---- corpus prob=  0.3749999999999999
W|->  W A: 0.125  W: 0.875
AY|->  I: 0.125  A I: 0.750  A: 0.125
N|->  N: 0.875  I N: 0.125
nonzeros: 7
```

```
iteration 3  ---- corpus prob=  0.6015625000000002
W|->  W A: 0.023  W: 0.977
AY|->  I: 0.023  A I: 0.955  A: 0.023
N|->  N: 0.977  I N: 0.023
nonzeros: 7


iteration 4  ---- corpus prob=  0.9126596543951916
W|->  W: 0.999
AY|->  A I: 0.999
N|->  N: 0.999
nonzeros: 3
AY : A I # 0.99889381
N : N # 0.99944690
W : W # 0.99944690
```

2. (5 pts) If you add `N AY N / N A I N` to it you should see it converge faster. Show the results.
   `N AY N / N A I N` would converge faster because as in the previous example, there are just three valid alignment combinations to begin with. Performing EM in this case would be much faster.
   Results from EM for this pair of eprons and jprons are as shown below:

```
iteration 0  ---- corpus prob=  0.0058593750000000035
N|->  N A: 0.167  N: 0.667  I N: 0.167
AY|->  I: 0.333  A I: 0.333  A: 0.333
nonzeros: 6


iteration 1  ---- corpus prob=  0.22222222222222227
N|->  N A: 0.083  N: 0.833  I N: 0.083
AY|->  I: 0.167  A I: 0.667  A: 0.167
nonzeros: 6


iteration 2  ---- corpus prob=  0.486111111111111
N|->  N A: 0.012  N: 0.976  I N: 0.012
AY|->  I: 0.024  A I: 0.952  A: 0.024
nonzeros: 6


iteration 3  ---- corpus prob=  0.9081227729186914
N|->  N: 1.000
AY|->  A I: 0.999
nonzeros: 2


iteration 4  ---- corpus prob=  0.9987817878961159
N|->  N: 1.000
AY|->  A I: 1.000
nonzeros: 2
AY : A I # 0.99999991
N : N # 0.99999995
```

3. (5 pts) Now that you have passed "wine", try the "test" example above (`T EH S T / T E S U T O`) and you'll see that EM converges to something wrong. Show me the result of 5 iterations.
   Result of running on "test" example is as given below:

```
iteration 0  ---- corpus prob=  0.0002603082049146184
T|->  T E S: 0.050  O: 0.300  T E: 0.150  T: 0.300  T O: 0.150  U T O: 0.050
EH|->  U: 0.100  S U: 0.100  E S U: 0.100  S: 0.200  E S: 0.200  E: 0.300
S|->  T: 0.300  U T: 0.200  S U T: 0.100  U: 0.200  S U: 0.100  S: 0.100
nonzeros: 18


iteration 1  ---- corpus prob=  0.0171
T|->  T E S: 0.013  O: 0.368  T E: 0.118  T: 0.368  T O: 0.118  U T O: 0.013
EH|->  U: 0.026  S U: 0.079  E S U: 0.158  S: 0.158  E S: 0.316  E: 0.263
S|->  T: 0.263  U T: 0.316  S U T: 0.158  U: 0.158  S U: 0.079  S: 0.026
nonzeros: 18


iteration 2  ---- corpus prob=  0.031396033640011964
T|->  O: 0.445  T E: 0.055  T: 0.445  T O: 0.055
EH|->  S U: 0.029  E S U: 0.180  S: 0.080  E S: 0.500  E: 0.210
S|->  T: 0.210  U T: 0.500  S U T: 0.180  U: 0.080  S U: 0.029
nonzeros: 14


iteration 3  ---- corpus prob=  0.06671740671405857
T|->  O: 0.491  T: 0.491
EH|->  E S U: 0.112  S: 0.015  E S: 0.757  E: 0.114
S|->  T: 0.114  U T: 0.757  S U T: 0.112  U: 0.015
nonzeros: 10
```

```
iteration 4  ---- corpus prob=  0.14474697012497167
T|->  O: 0.500  T: 0.500
EH|->  E S U: 0.021  E S: 0.957  E: 0.021
S|->  T: 0.021  U T: 0.957  S U T: 0.021
nonzeros: 8
EH : E # 0.02121672
EH : E S # 0.95723661
EH : E S U # 0.02120941
S : S U T # 0.02120941
S : T # 0.02121672
S : U T # 0.95723661
T : O # 0.49983138
T : T # 0.49983138
```

Now come up with a minimal second example so that when combined with the "test" example, EM will learn the correct alignments. This second pair has to be a real English word, and should be choosen from `epron-jpron.data`.

4. (8 pts) Now come up with an example dataset where EM does not learn anything and is still ambivalent at the end (i.e., the final probability table is (almost) the same as the initial one). In the lecture we gave `B IY / B I I` as one such example, but can you come up with something bigger, i.e., longer sequences and multiple, interdependent pairs? **Another example dataset where EM is still ambivalent is** `IH CH /I TCH I`
Output from EM for given epron-jpron pair is as follows:

```
iteration 0  ---- corpus prob=  0.08000000000000002
IH|->  I TCH: 0.500  I: 0.500
CH|->  I: 0.500  TCH I: 0.500
nonzeros: 4


iteration 1  ---- corpus prob=  0.5
IH|->  I TCH: 0.500  I: 0.500
CH|->  I: 0.500  TCH I: 0.500
nonzeros: 4


iteration 2  ---- corpus prob=  0.5
IH|->  I TCH: 0.500  I: 0.500
CH|->  I: 0.500  TCH I: 0.500
nonzeros: 4


iteration 3  ---- corpus prob=  0.5
IH|->  I TCH: 0.500  I: 0.500
CH|->  I: 0.500  TCH I: 0.500
nonzeros: 4


iteration 4  ---- corpus prob=  0.5
IH|->  I TCH: 0.500  I: 0.500
CH|->  I: 0.500  TCH I: 0.500
nonzeros: 4
CH : I # 0.50000000
CH : TCH I # 0.50000000
IH : I # 0.50000000
IH : I TCH # 0.50000000
```

5. (15 pts) If you have passed all previous questions, it's now time to work on the whole dataset with the following command-line:

```
cat epron-jpron.data | ./em.py 15 >epron-jpron.probs 2>epron-jpron.logs
```

(Ignore the alignment lines in the input – your job is to learn them yourself!)

The logs file contains the logging info for all iterations (see above), and print the final prob. table to the `.probs` file. Theis file should follow the same format as in HW3 (ignore < 0.01 entries). **The epron-jpron.probs and corresponding epron-jpron.logs file have been included in the zip.**

Note: Do not reestimate probs from the final Viterbi alignments – you'll get HW3 probs that way. **Hint:** The corpus probs are likely to underflow (beyond the range of Python `float`). Try fix it.

6. (7 pts) Decode `jprons.txt` from HW3 using your newly learned `epron-jpron.probs`. Compare with your results from HW3: did you see any differences?

The results from decoding jprons.txt by using epron-jpron.probs generated using EM and kbest.py from HW3 is as given below:

```
        HW3 decoding                                               HW4 decoding
HILARY CLINTON 7.88860055085763e-15                     | HILARY CLINTON 8.03696552254933e-15
DONALD TRUMP 6.07339610594789e-14                       | DONALD TRUMP 6.35715029452527e-14
VIDEOTAPE 7.45026118607828e-09                          | VIDEOTAPE 5.63071802159069e-09
HOMER SIMPSON 4.36333153281140e-16                      | HOMER SIMPSON 4.39708714769990e-16
LAPTOP 2.53339181062686e-08                             | LAPTOP 1.55670383757238e-08
SAVING CREAM 8.94914640410037e-13                       | SAVING CREAM 7.45513900549216e-14
CHILD SHEET 2.21259613170082e-11                        | CHILD SHEET 2.34543785345353e-11
SEAT BELT 1.20997626367931e-11                          | SHEET BELT 3.31933767422258e-12
SINGLE ROOM 3.74244012372088e-11                        | SINGLE ROOM 2.79834799846408e-12
```

```
GIRLFRIEND 6.71285977991981e-08                   | GIRLFRIEND 6.94250861316719e-08
TRAVELERS CHECK 2.01766655996498e-14              | TRAVELERS CHECK 2.14644530061084e-14
BABYSITTER 6.55532343593239e-11                   | BABYSITTER 5.40924616208129e-11
SCOTLAND 2.66088826702559e-08                     | SCOTLAND 2.70448084776106e-08
VIOLIN CONCERTO 1.07026890193206e-17              | VIOLIN CONCERTO 8.74070445144713e-18
APPLE MAKE BOOK PRO 1.84707628147010e-22          | APPLE MAKE BOOK PRO 1.97776200703538e-22
COMPUTER SCIENCE 3.64010964458249e-14             | COMPUTER SCIENCE 1.78059934635410e-14
PHYSICAL TRAINING 2.85412778532932e-13            | PHYSICAL TRAINING 2.92093712224797e-13
PHYSICAL EXERCISE 1.67630338017214e-15            | PHYSICAL EXERCISE 1.68032933046351e-15
ICE CREAM 4.47342795007371e-12                    | ICE CREAM 4.53176428844176e-12
HOT MILK 8.30222711400697e-12                     | VOTE MILK 3.26886729019331e-11
TRIPLE ROOM 2.41054062684565e-12                  | TRIPLE ROOM 2.21502904440440e-12
CROWN PLAZA HOTEL 2.51647416168438e-18            | CROWN PLAZA HOTEL 2.52883818491349e-18
FACE BOOK RESEARCH SCIENTIST 4.34881845427921e-22 | FACE BOOK RESEARCH SCIENTIST 4.66746013149445e-22
WOLFGANG MOZART 6.91333429384525e-21              | WOLFGANG MOZART 3.15556703913702e-20

Probabilities are different, but all ewords are the same except for:
    SEAT BELT 1.20997626367931e-11                    | SHEET BELT 3.31933767422258e-12
    HOT MILK 8.30222711400697e-12                     | VOTE MILK 3.26886729019331e-11
```

7. (5 pts) Generate Viterbi alignments from `epron-jpron.probs`:

```
cat epron-jpron.data | ./viterbi_align.py epron-jpron.probs >epron-jpron.viterbi
```

The result file should follow the same format as `epron-jpron.data` so that we can compare them.
Viterbi-alignments generated using epron-jpron.probs from EM is attached with the zip.

8. (5 pts) Calculate the number of different viterbi alignments between your `epron-jpron.viterbi` and `epron-jpron.data`. It should be less than 10 (out of 2684 examples).
The number of different Viterbi alignments are 5.

4 EM, Implementation II: Forward-Backward (DP) (30 pts)

1. (5 pts) **BTW how many possible alignments are there for a pair of n English phonemes and m Japanese phonemes ($n \leq m$)?**

   Number of possible alignments can be as described using following formula:

   $$n * (\binom{m}{1} + \binom{m}{2} + \binom{m}{3})$$

   where 1,2,3 correspond to respective number of Japanese phonemes selected at a time.

   What pair(s) in `epron-jpron.data` give the highest number of alignments?

   pairs where m is largest, and $n \approx \frac{m}{2}$. For example:

   ```
   SH UW T
   SH Y U U T O
   ```

2. (20 pts) **Implement the forward-backward algorithm Make sure the two approaches match on their results.**
   Yes, they match.

   Include a detailed pseudocode of your implementation.

   - initialize the conditional probabilities `pDict` for every valid alignment of epron-jpron pair to uniform. (Valid meaning each epron corresponding to combination of one to maximum three Japanese phonemes.)
   - repeat for given number of iterations:
   - **E-step**:
   - Initialize corpus probability to 0 and countDict to store the fractional counts.
   - **Perform Forward Viterbi**:
   - For each English phoneme epron[i]:
   - For all valid corresponding Japanese phoneme sequences jpron of lengths k = 1 to 3:

- Calculate and collect forward probabilities fwd[i+1][j+k] = fwd[i][j] * pDict[epron][jpron] (j is the index of the beginning of jpron, k = 1,2,3)
- Save the total forward probability at the last stage as FTprob.
- **Perform Backward Viterbi**:
- For each English phoneme epron[i]:
- For all valid corresponding Japanese phoneme sequences (starting in reverse from the last Japanese phoneme) jpron of lengths k = 1 to 3:
- Calculate and collect backward probabilities bck[i][j-k] = bck[i+1][j+1] * pDict[epron][jpron] (j+1 is the index of the beginning of jpron, k = 1,2,3)
- **(collect the fractional counts)** For every English-Japanese phoneme sequence pair:
- Store the fractional counts in countDict[epron][jpron] += pDict[epron][jpron] * fwd[e][j] * bck[e+1][j+k] / FTprob
- **M-step**:
- count-and-divide based on the collected fractional counts from all pairs to get new prob. table until reached maximum iteration or converged

- Normalize countDict and store in pDict after eliminating the pairs for which probabilities ≤ 0.01

What is the complexity of this forward-backward algorithm for a (n, m) pair?

$\mathcal{O}(n * m * k)$ where k = 1,2,3
In general form, the complexity of this forward-backward algorithm will be number of english phonemes multiplied by the number of japanese phonemes. More specifically, the complexity in the forward step (also in backward) will vary depending on value of length k (1,2 or 3).

How does it compare to the enumeration approach in terms of speed?

The enumeration approach is slower than forward-backward algorithm. Enumeration approach takes more time as it goes over all possible alignments, whereas forward-backward takes into account the best probabilistic score until the ith symbol, both in the forward as well as the backward direction.

3. (5 pts) **Try to construct an example so that `em2.py` is substantially faster than** `em.py| and explain why.`
The example is:

```
EH R EY S B AE S K
E E R U E E S U B A S U K U
```

```
time to complete 10 iterations DP:
TotalSeconds      : 0.1544489

time to complete 10 iterations non-DP:
TotalSeconds      : 50.3837504
```

This takes the em algorithm a VERY long time to compute, because there are a huge number of possible alignments with 8 eprons going to 14 jprons. The forward-backward algorithm is very quick. (note: I just combined multiple lines from epron-jpron.data to create this example. it is not a single english word.)

5 EM for Decipherment (30 pts)

Can you decode this? Hint: letter substitution cipher (i.e., permutation of 27 chars).

```
gjkgcbkycnjpkovryrbgkcrvsczbkyhkahqvykgjvjkcpekrbkjxdjayrpmkyhkmhkyhkyvrcukrpkbjfjvcukihpygb
oqykcykujcbykhpjkejihavcyrakvjmrijkjxrbybkrpkjfjvzkiclhvkarfrurtcyrhpkhvkaquyqvjkrpkygjkshvue
auqobkrpkvjajpykzjcvbkgcfjkwhqpekohvcbkbhkerwwraquykyhkpjmhyrcyjksrygkygcykicpzkbgqpkgrbkaurjpyb
gjkbcrekygjkoqvjcqkiqbykgcfjkygjkerbdqyjkvjbhufjekozkwjovqcvzkrpkhvejvkyhkdjvwhvikygjkajpbqb
oqykygjkdqouraryzkbqvvhqperpmkygjkejocaujkajvycrpuzkbjvfjekyhkwhaqbkckbdhyurmgykhpkyghbjkjwwhvyb
```

(Wait: how come there is no space at all, given that space also participated in the permutation??)

Using a bi-gram model trained on Ex2's `train.txt` with 50 iterations should be good enough.
Hint: as shown in slides, start with a uniform model $p(c \mid e) = 1/27$ for all c, e: (the LM does not change!)

$$\begin{aligned} p(c_1 \dots c_n) &= \sum_{e_1 \dots e_n} p(c_1 \dots c_n, e_1 \dots e_n) = \sum_{e_1 \dots e_n} p(e_1 \dots e_n) \cdot p(c_1 \dots c_n \mid e_1 \dots e_n) \quad (1) \\ &= \sum_{e_1 \dots e_n} p(e_1 \dots e_n) \cdot p(c_1 \mid e_1) \cdots p(c_n \mid e_n) \quad (2) \\ &= \sum_{e_1 \dots e_n} p(e_1 \mid \texttt{<s>}) \cdots p(e_i \mid e_{i-1}) \cdots p(\texttt{</s>} \mid e_n) \cdot p(c_1 \mid e_1) \cdots p(c_n \mid e_n) \quad (3) \end{aligned}$$

After each iteration, print some debugging information like these (probs ≤ 0.01 are considered zeros):

```
$ cat cipher.txt | ./decipher2.py train.txt 50
epoch   1 logp(corpus)= -2270.05  entropy= 4.79 nonzeros= 567
...
epoch  50 logp(corpus)= -1606.14  entropy= 3.39 nonzeros= 76
```

notice the initial entropy is similar to 0-gram's, and the final entropy similar to 2-gram's (see LM slides). why?

5.1 cipher3

```
fnwmwyrpxqwqmucqwemgwsfmxbmonwsfmrqxfwoftxbmuxqmgxfnmekjjtws
fnwmlxcpmtsmfxmewfwqjtbwmnx mvwntopwsmocbmgwmqwewstlbwemfxmjciwmocqmxookrcbfsmscuwqm tfnxkfmjcitblmptlni
gkfmnwmsctemukqfnwqmqwswcqonm csmbwwewemfxmewfwqjtbwmfnwmqwpcftvwmtjrxqfcbowmxumfnwswmtsskwsmcbem
nwfnwqmcbzmbw mqwlkpcftxbsmjtlnfmgwmbwwewe
jcqftbwamcbemckfxmtbeksfqzmxuutotcpsmsrxiwmcfmcbmtbfwqbcftxbcpmfqcuutomscuwfzmoxbuwqwbowmtbmfntsmucofxq
bmcoqxssmfnwmewfqxtfmqtvwqmuqxjmewfqxtf
qwlkpcfxqsmsctemfnwzmncemwyrwofwemfnwmocqmfxmtbuptofmpwssmecjclw
```

cipher3.deciphered Accuracy - 91.94%

```
the explorer fared best on chest protection for both bummies
the coal is to determine how vericles can be redesicked to make car ongupants waver without
making licht truck ongupants less wave
but he wand jurther research was beeded to determine the relative importange of these issues
and whether any bed requlations micht be beeded
martined and auto industry officialy spoke at an international thaffin wavety congerenge in
this fantory town across the dethont river from dethont
requlators wand they had expected the car to inglint less bamace
```

5.2 cipher4

```
dlktajetakbxxtaeeatptabxzectxbwbwqtbwtkjetfjyxetrbqjk
bktveckpbwxutaeehatkytjpzetdeewthycetzbqycylatbwtkjetnpak
aezecpxtzbabkycatapbitkjeutfecetibapnnybwkeitbwtkjetklcwylk
qcpnjbvthpntbatdebwqtaewktkytwuktqcpnjbvtvxbewka
cenldxbvpwtapbitbwtpwtehykbywpxtaneevjtywtkjetaewpketrxyyc
iedpketytwtjetydpvvytdbxxtbatavjeilxeitkytcealhetkjlcaipu
bt weftkjeutfecetajyfbwqtptxyktyrtbwkeceak
jetapbitjbatnxpwtfpatkytabqwtpatolbv xutpatnyaabdxe
jetjpatkjetpdbxbkutkytnxputbwtkjetdbqtxepqleatayyw
npcktyrtkjetibrrecewvetbatjyftpnncevbpkbywtbatvpxvlxpkei
```

cipher4.deciphered Accuracy - 92.42%

```
but she still sees a silver lining in the whole fight
it certainly seers to have been more wigorous in the past
several wisitors wand they were disappointed in the turnout
grathid mat is being sent toun t grathid ckients
republican wand in an erotional speech on the senate floor
```

```
demate oun he omacco will is schequled touresure thershay
i aned they were shofing a lot of interest
he wand his plan was to sign as juicaly as possicke
he has the amility to play in the wig leabues soon
part of the differende is hof apprediation is caldulated
```

5.3 cipher5

```
remxahbxamqssxabbaxjxaqscbtxsququgxquxmhbxfhisbxlqghm
qmxpbtmjquszxabbvaxmixhjcbxrbbuxvitbxcqgitieaxquxmhbxwjam
abcbtjsxcqaqmitaxajqyxmhbzxfbtbxyqajwwiqumbyxquxmhbxmetuiem
gtjwhqpxvjwxqaxrbqugxabumxmixuzmxgtjwhqpxpsqbuma
tbwersqpjuxajqyxquxjuxbvimqiujsxawbbphxiuxmhbxabujmbxlsiit
ybrjmbxiuxmhbxmirjppixrqssxqaxaphbyesbyxmixtbaevbxmhetayjz
qxnubfxmhbzxfbtbxahifqugxjxsimxilxqumbtbam
hbxajqyxhqaxwsjuxfjaxmixaqguxjaxdeqpnszxjaxwiaaqrsb
hbxhjaxmhbxjrqsqmzxmixwsjzxquxmhbxrqgxsbjgebaxaiiu
wjtmxilxmhbxyqllbtbupbxqaxhifxjwwtbpqjmqiuxqaxpjspesjmby
```

cipher5.deciphered Accuracy - 92.8%

```
but whe still sees a silver lining in the whole fight
it certainly seers to have been more wigorous in the past
several wisitors wand they were disappointed in the turnout
grathid mat is being sent to nst grathid ckients
republican wand in an erotional speech on the senate floor
demate on the tomacco will is schequled to resure thers ay
i aned they were whofing a lot of interest
he wand his plan was to sign as quicaly as possicke
he has the amility to play in the wig leabues soon
part of the differende is hof apprediation is caldulated
```

5.4 cipher6

```
gjkgcbkycnjpkovryrbgkcrvsczbkyhkahqvykgjvjkcpekrbkjxdjayrpmkyhkmhkyhkyvrcukrpkbjfjvcukihpygb
oqykcykujcbykhpjkejihavcyrakvjmrijkjxrbybkrpkjfjvzkiclhvkarfrurtcyrhpkhvkaquyqvjkrpkygjkshvue
auqobkrpkvjajpykzjcvbkgcfjkwhqpekohvcbkbhkerwwraquykyhkpjmhyrcyjksrygkygcykicpzkbgqpkgrbkaurjpyb
gjkbcrekygjkoqvjcqkiqbykgcfjkygjkerbdqyjkvjbhufjekozkwjovqcvzkrpkhvejvkyhkdjvwhvikygjkajpbqb
oqykygjkdqouraryzkbqvvhqperpmkygjkejocaujkajvycrpuzkbjvfjekyhkwhaqbkckbdhyurmgykhpkyghbjkjwwhvyb
```

cipher6.deciphered Accuracy - 86.5%

```
he has taven pritish aingays to fourt here and is ewhextind to co to trial in several monthe
but at least one demofratif redime ewists in every macor fivilization or bullure in the world
blups in refeng gears have cound poras so wickibull to necotiate with that mang then his blients
he waid the bureay must have the wisqute resolved by cexpuary in onder to hercory the bensus
but the quplifity suproundind the depable bertainly served to cofus a thollicht on those eccorts
```

5.5 cipher9

```
vskbnmkdegskpnvskneiqnhivsnmtcvuiqsvmnoaviunvskbnrcoapndapnekmgokpnvsknyimmiaqnhcewke
vskbnakxkenvscoqsvnd covndabnpdaqkenvcnvskymkuxkmnoaviunivnhdmnduuncxke
vskbntdipnvei ovknvcnvskyndnbkdendqcnhivsndnpiaakenianscomvca
sknsdmnvdwkan eivimsndiehdbmnvcngcoevnskekndapnimnkjtkgviaqnvcnqcnvcnveidunianmkxkedunycavsm
 ovnmcykncovmipkemnmdbnvsknriqsvnimndmnyogsnrcento uigndvvkavicandmnivnimnrcenukqdunxigvceb
 ovnmsknmviuunmkkmndnmiuxkenuiaiaqnianvsknhscuknriqsv
vsknmiqamndeknteiavkpnianekxkemkndapndeknykdavnvcn knekdpnianvsknyieece
 ovndvnukdmvncaknpkycgedvignekqiyknkjimvmniankxkebnydzcengixiuildvicancengouvoeknianvsknhceup
vskbngdantoemokndqqekmmixknadvicaduimvigntcuigikmndapntoemokneoiacomnkgcacyigntcuigb
vsimngdyknvsecoqsntdevigoudeubngukdeniandntdtkentektdekpn bndnmgscudenrecyniapid
ivngkevdiaubnmkkymnvcnsdxkn kkanyceknxiqcecomnianvskntdmv
 ovnrkhngcamkexdvixknkoucqimvmnsdxknykavicakpnvskmknhdeaiaqmnvsimnhkkw
```

```
vsimnektekmkavmnvskngkavedunpedydncrnektecpogvixknskduvsniandreigd
yimmndwdnmtkapmnyogsncrnskenviyknekqekvviaqnvsdvnmsknsdpnacvn kkan kvvkeniarceykpnd covnskengscigkmn
krceknskenmkgcapntekqadagb
qccpydanmdipntkctuknmscoupnmvdapnotnvcnvsknvd ucipmnhivsnyceknxiqce
 dudagsiaknmtcvvkpnskend iuivbnvcndadublkndapngceekgvnhskanmsknhdmnianvskngcetmnpkn duukv
msknhdmnuiwkndnuivvukngcmyigntdeviguknruivviaqndecoapnvsknmvdqk
skenvkdgsiaqngcaviaokmnvcnkagcytdmmnukgvoeknpkycamvedvicamndapnvsknpimmkyiadvicancrnvskn dudagsiaknkvsc
ecdp
deilcadndvvceakbnqkakedunqedavnhccpmnsdpnmcyknekqekvmndrvkentdmmiaqn bnvskn ike
mkxkedunximivcemnmdipnvskbnhkeknpimdttciavkpnianvsknvoeacov
```

cipher9.deciphered Accuracy - 95.94%

```
they searthed the rig with spollights until they found and rescked the missing worver
they never thought about any danger to themselves until it was all over
they paid tribute to them a year ago with a dinner in houston
he has taven british airmays to court here and is expecting to go to trial in several monthe
but some outsiders way the fight is as quch for jublin athention as it is for lecal wictory
but whe still sees a silver lining in the whole fight
the signs are printed in reverse and are meant to be read in the mirror
but at least one democratin regime exists in every macor civilization or cullure in the world
they can jurske angressive mationalistin policies and jurske rkinous economin policy
this came through particularly ckear in a paper prepared by a stholar from india
it certainly seems to have been more wicorous in the past
but few conservative aulogists have mentioned these warnings this weev
this represents the central drama of reproductive health in africa
miss ala spends quch of her time regrething that whe had fot been bether informed about her
thoices before her second pregnancy
goodman waid pexple whould stand up to the tabloids with more wicor
balanthine spothed her ability to analyze and correct when whe was in the corps de ballet
whe was live a litthe cosmin particke flithing around the stage
her teathing continkes to encompass lecture demonstrations and the dissemination of the balanthine
athos to teathers abrond
arizona athorney general grant woods had some regrets acher passing by the bier
several wisitors waid they were disappointed in the turnout
```

Optional: Can you do trigram? (should be better: the stronger your LM is, the better you can decipher).

Optional: What if you also update the language model during EM training (like in Eisner's example)?