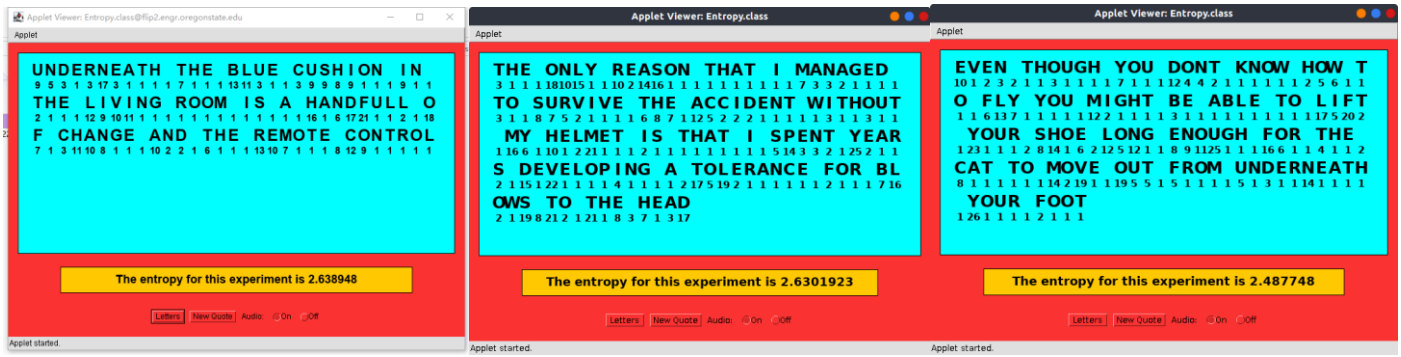
NLP HW2

1. Shannon Game and Entropy of English



→ The entropy was calculated in a classic way. The exact code for the entropy calculation is:

```
def entropy(guess_seq):
    s = np.array(Counter(guess_seq).values()) / len(guess_seq)
    return -sum( p * math.log(p, 2) for p in s)
```

2. Part-of Speech Tagging as WFST Composition

3) The results of tag sequences from Carmel output

cat part2_3.txt | ./carmel -slibOEQk 5 lexicon.wfst bigram.wfsa
Input line 1: I hope that this works
(6 states / 13 arcs)
(15 states / 18 arcs reduce-> 9/10)
PRO V CONJ PRO V 2.643354e-05
N V CONJ PRO V 7.56000000000003e-09
N N CONJ PRO V 7.56e-10
0
0
Input line 2: They can fish
(4 states / 6 arcs)
(9 states / 10 arcs reduce-> 7/7)
PRO AUX V 0.002016
PRO V N 0.000945
0
0
0
Input line 3: A panda eats shoots and leaves
(7 states / 8 arcs)
(12 states / 12 arcs reduce-> 9/9)
DET N V N CONJ N 5.0625e-05
DET N V N CONJ V 1.6875e-05
0
0

0
Input line 4: They can can a can
(6 states / 11 arcs)
(12 states / 13 arcs reduce-> 7/6)
PRO AUX V DET N 2.268e-05
0
0
0
0
Input line 5: Time flies like an arrow
(6 states / 14 arcs)
(14 states / 20 arcs reduce-> 10/12)
N V PREP DET N 7.08750000000001e-05
N V CONJ DET N 5.66999999999999e-05
N N CONJ DET N 2.268e-05
N N V DET N 1.18125e-05
0
Derivations found for all 5 inputs
Viterbi (best path) product of probs=4.3365765216429e-21, probability=2^-67.6439 per-input-symbol-perplexity(N=24)=2^2.8185 per-line-perplexity(N=5)=2^13.5288

4) Pipeline mathematically sound?

→ **Taking sentence "A panda eats shoots and leaves" as an example, the highest probability for this sentence is DET N V N CONJ N 5.0625e-05, and the calculation is showing below. Obviously, the pipeline probability we calculated by hand is equal to the probability given by camrel. Therefore, it is mathematically sound.**

$$(P(DET|0) \times P(DET|A)) \times (P(N|DET) \times P(N|panda)) \times (P(V|N) \cdot P(V|eats)) \times (P(N|V) \times P(N|shoots)) \times (P(CONJ|N) \times P(CONJ|and)) \times (P(N|CONJ) \times P(N|panda)) \times P(END|N)$$

$$= (0.2 \times 1) \times (1 \times 1) \times (0.25 \times 1) \times (0.3 \times 0.5) \times (0.5 \times 1) \times (0.2 \times 0.5) \times 0.45$$

$$= 0.000050625 = 5.0625e-05$$

5) funny interpretation, how to fix them?

Below is the result from bigram_old.wfsa without probabilities:

Input line 3: Time flies like an arrow
(6 states / 14 arcs)
(14 states / 20 arcs reduce-> 10/12)
N N V DET N
N V PREP DET N
N N CONJ DET N
N V CONJ DET N

→ The original intention of this sentence is the second result, but it is not ranked at the top. In order to fix it, we added more reasonable probability to bigram.wfsa. Then, the second result jumped to first place and showed the correct tag of the sentence.

3. Pronouncing and Spelling English

1) Words: DAVID, EXAMPLE, COMPUTER, GUIDELINES, OCTOBER

```
cat word3_1.txt | ./carmel -slibOEQk 5 eword-epron.wfst
Input line 1: DAVID
        (209214 states / 209213 arcs reduce-> 9/8)
D EY V IH D 1
0
0
0
0
Input line 2: EXAMPLE
        (209214 states / 209213 arcs reduce-> 12/11)
IH G Z AE M P AH L 1
0
0
0
0
Input line 3: COMPUTER
        (209214 states / 209213 arcs reduce-> 12/11)
K AH M P Y UW T ER 1
0
0
0
0
Input line 4: GUIDELINES
        (209214 states / 209213 arcs reduce-> 11/10)
G AY D L AY N Z 1
0
0
0
0
Input line 5: OCTOBER
        (209214 states / 209213 arcs reduce-> 10/9)
AA K T OW B ER 1
0
0
0
0
Derivations found for all 5 inputs
Viterbi (best path) product of probs=1, probability=2^0 per-input-symbol-perplexity(N=5)=2^-0 per-line-perplexity(N=5)=2^-0
```

2) Words: CAMOUFLAGE, PEPPER, MACHINE, LANGUAGE, PREDICTIVE

cat word3_2.txt | ./carmel -sribIEQk 5 epron-espell.wfst
Input line 1: C A M O U F L A G E
(768005 states / 768627 arcs reduce-> 1405/2027)
IY S IY T R IH EH M OW Y AH F L IH JH 1
IY S IY T R IH EH M OW Y AH F AH L AE JH IY IY 1
IY S IY T R IH EH M OW Y AH W F EH B Y L IH JH 1
IY S IY T R IH EH M OW Y AH F AH L AE IY JH IY IY S T 1
IY S IY T R IH EH M OW Y AH W F EH B Y AH L AE JH IY IY 1
Input line 2: P E P P E R
(486547 states / 486889 arcs reduce-> 782/1124)
P EH P IY AA R 1
P EH P AH N UW IH S IH 1
P Y AH N UW P Y Z P ER IY AA R 1
P EH P AH N UW N Y ER 1
P Y AH N UW P Y Z P ER AH N UW IH S IH 1
Input line 3: M A C H I N E
(557634 states / 558146 arcs reduce-> 1213/1725)
M EY S HH IH N AH N UW 1
M EY S HH IH N IY S T 1
M EY S HH IH EH N EH AH N UW 1
AH M EH T IY AH T ER AH HH IH N AH N UW 1
M EY S HH IH EH N EH IY S T 1
Input line 4: L A N G U A G E
(628345 states / 628906 arcs reduce-> 1380/1941)
L AA NG G Y UW EY JH IY IY 1
L AA NG G Y UW EY IY JH IY IY S T 1
L AA NG G AH W EY JH IY IY 1
L AA NG G Y UW EY IY JH IY AH N UW 1
L AA NG G AH W EY IY JH IY IY S T 1
Input line 5: P R E D I C T I V E
(764678 states / 765319 arcs reduce-> 1441/2082)
P R EH D IH K T IY EH IH V AH 1
P R EH D IH K T IY EH IY AY V AH N UW 1
P R EH D IH K T IH V AH N UW 1
P R EH D IH K T IY EH IY AY IY V IY AH N UW 1
P R EH D IH K T IH V IY S T 1
Derivations found for all 5 inputs
Viterbi (best path) product of probs=1, probability=2^0 per-input-symbol-perplexity(N=41)=2^-0 per-line-perplexity(N=5)=2^-0

→ There are lots of nonsense outputs because the majority of output is based on each letter, and the pronunciation of each letter will not consider the letters before and after it. Therefore, we could find that most outputs are the composition of each letter's highest probability pronunciation among its all possible combinations.

3) Words: CAMOUFLAGE, SERENDIPITY, BISCUITS, DERIVATIVES, LANGUAGE

cat word3_3.txt | ./carmel -sribIEQk 5 epron.wfsa epron-espell.wfst
Input line 1: C A M O U F L A G E

(768005 states / 768627 arcs reduce-> 1405/2027)
(17274 states / 167444 arcs reduce-> 14622/151523)
K AA M OW F L IH JH 1.15268956155813e-13
K AH M OW F L IH JH 1.15220423647597e-13
K AA M OW F L IH JH 9.83319285592901e-14
K AA M OW F L IH JH 7.97943292146739e-14
K AH M OW F L IH JH 7.97607328408762e-14
Input line 2: S E R E N D I P I T Y
(834155 states / 834863 arcs reduce-> 1589/2297)
(18263 states / 188905 arcs reduce-> 16078/172144)
S ER EH N D IH P AH T IY 1.23008080029545e-12
S ER EH N D IH P AH T IY 8.65243546614445e-13
S ER EH N D IH P AH T IY 7.56027728998865e-13
S ER EH N D IH P AH T IY 7.25672245094909e-13
Z ER EH N D IH P AH T IY 6.40612973281397e-13
Input line 3: B I S C U I T S
(625125 states / 625633 arcs reduce-> 1103/1611)
(14537 states / 169470 arcs reduce-> 12668/155887)
B IH S K UW T S 3.04987687866664e-10
B IH S K UW T S 1.79385386135482e-10
B IH S K UW T S 9.27460733080474e-11
B IH S K AH T S 3.38396679331213e-11
B IH S K UW T S 2.21222303526983e-11
Input line 4: D E R I V A T I V E S
(835265 states / 836037 arcs reduce-> 1806/2578)
(20402 states / 199836 arcs reduce-> 17698/181072)
D EH R AH V EY T IH V Z 4.85248805445689e-12
D EH R AH V EY T IH V Z 4.59752696620892e-12
D EH R IY V AH T IH V Z 4.38807442796781e-12
D EH R IY V AH T IH V Z 4.1575147194405e-12
D EH R AH V EY T IH V Z 3.97327370224058e-12
Input line 5: L A N G U A G E
(628345 states / 628906 arcs reduce-> 1380/1941)
(15924 states / 167432 arcs reduce-> 13287/152738)
L AH N G EY JH 2.68881799191052e-09
L AE N G EY JH 2.27820212901568e-09
L AE N G EY JH 2.13703348316821e-09
L AH N G EY JH 1.05714000682781e-09
L AA N G EY JH 7.90028379899598e-10
Derivations found for all 5 inputs
Viterbi (best path) product of probs=e^-124.911892902887, probability=2^-180.21 per-input-symbol-perplexity(N=48)=2^3.75437 per-line-perplexity(N=5)=2^36.042

4) Prons: T ER M IY N AH L, F EY N AE T IH K, AA K S F ER D, EH M P L OY IY

cat pron3_4.txt | ./carmel -slibOIEQk 50 epron-espell.wfst > **part3_4.txt**
Input line 1: T ER M IY N AH L
(8575 states / 9602 arcs reduce-> 2860/3887)
Input line 2: F EY N AE T IH K

(8025 states / 8807 arcs reduce-> 2078/2860)
Input line 3: AA K S F ER D
(7055 states / 7734 arcs reduce-> 1877/2556)
Input line 4: EH M P L OY IY
(5641 states / 6132 arcs reduce-> 1264/1755)
Derivations found for all 4 inputs
Viterbi (best path) product of probs=0.20226001619897, probability=2^-2.30572 per-input-symbol-perplexity(N=26)=2^0.0886814 per-line-perplexity(N=4)=2^0.576429

5) Prons: T ER M IY N AH L, F EY N AE T IH K, AA K S F ER D, EH M P L OY IY

cat pron3_4.txt | ./carmel -slibOIEQk 50 epron-espell.wfst espell-eword.wfst eword.wfsa > **part3_5.txt**
Input line 1: T ER M IY N AH L
(8575 states / 9602 arcs reduce-> 2860/3887)
(2610466 states / 6390081 arcs reduce-> 122257/2432452)
(120276 states / 1906664 arcs reduce-> 42268/1518927)
Input line 2: F EY N AE T IH K
(8025 states / 8807 arcs reduce-> 2078/2860)
(1863363 states / 4603999 arcs reduce-> 67286/1743543)
(66908 states / 1407557 arcs reduce-> 28719/1132562)
Input line 3: AA K S F ER D
(7055 states / 7734 arcs reduce-> 1877/2556)
(1643006 states / 4646247 arcs reduce-> 70094/1907578)
(67896 states / 1671147 arcs reduce-> 29335/1332700)
Input line 4: EH M P L OY IY
(5641 states / 6132 arcs reduce-> 1264/1755)
(178614 states / 1887603 arcs reduce-> 11869/1159365)
(11348 states / 1034193 arcs reduce-> 4680/690981)
Derivations found for all 4 inputs
Viterbi (best path) product of probs=2.28470467793547e-22, probability=2^-71.8904 per-input-symbol-perplexity(N=26)=2^2.76502 per-line-perplexity(N=4)=2^17.9726

6) It is not better than 5).

cat pron3_4.txt | ./carmel -slibOEQk 50 epron-eword.wfst eword.wfsa > **part3_6.txt**
Input line 1: T ER M IY N AH L
(40 states / 49 arcs reduce-> 17/18)
(16 states / 16 arcs)
Input line 2: F EY N AE T IH K
(27 states / 31 arcs reduce-> 15/16)
(14 states / 14 arcs)
Input line 3: AA K S F ER D
(25 states / 28 arcs reduce-> 10/9)
(10 states / 9 arcs)
Input line 4: EH M P L OY IY
(20 states / 21 arcs)
(20 states / 21 arcs)
Derivations found for all 4 inputs
Viterbi (best path) product of probs=e^-84.0470893657553, probability=2^-121.254 per-input-symbol-perplexity(N=26)=2^4.66363 per-line-perplexity(N=4)=2^30.3136

7) Words: NURBURGRING, RAIKKONEN, NANOTECH, CYBERBULLY

```
cat word3_7.txt | ./carmel -slibOEQk 5 eword-epron.wfst
Input line 1: NURBURGRING
        (0 states / 0 arcs)
Empty or invalid result of composition with transducer "eword-epron.wfst".
0
0
0
0
0
Input line 2: RAIKKONEN
        (0 states / 0 arcs)
Empty or invalid result of composition with transducer "eword-epron.wfst".
0
0
0
0
0
Input line 3: NANOTECH
        (0 states / 0 arcs)
Empty or invalid result of composition with transducer "eword-epron.wfst".
0
0
0
0
0
Input line 4: CYBERBULLY
        (0 states / 0 arcs)
Empty or invalid result of composition with transducer "eword-epron.wfst".
0
0
0
0
0
No derivations found for 4 of 4 inputs
Viterbi (best path) product of probs=1, probability=2^0 per-input-symbol-perplexity(N=4)=2^-0, excluding 4 0
probabilities (i.e. real ppx is infinite).
```

8) Words: NURBURGRING, RAIKKONEN, NANOTECH, CYBERBULLY

```
cat word3_8.txt | ./carmel -sribIEQk 5 epron.wfsa epron-espell.wfst
Input line 1: N U R B U R G R I N G
        (834361 states / 834946 arcs reduce-> 1399/1984)
        (15388 states / 143298 arcs reduce-> 13197/129159)
N ER B ER G R IH NG 2.46739103853511e-11
N ER B ER G R IH NG 1.12249614468061e-11
N ER B ER G R IH NG 1.06351724866701e-11
N ER B ER G R IH NG 3.31928793585288e-12
N ER B ER G ER IH NG 2.86617708735532e-12
```

Input line 2: R A I K K O N E N
(696437 states / 696980 arcs reduce-> 1210/1753)
(14130 states / 129834 arcs reduce-> 12300/116713)
R EY IH K AH N AH N 1.28224720572323e-13
R EY IH K AH N AH N 6.54392133843099e-14
R EY K AH N AH N 5.67674352563216e-14
R EY IH K AH N AH N 5.61594590183819e-14
R AA IY K OW N AH N 5.34161180316217e-14
Input line 3: N A N O T E C H
(627439 states / 627986 arcs reduce-> 1208/1755)
(16765 states / 205765 arcs reduce-> 14510/189978)
N AA N OW T EH K 2.1142951137977e-09
N AA N OW T EH K 1.97734683478091e-09
N AA N OW T EH K 1.15145484367419e-09
N AA N OW T EH K 1.00309877907757e-09
N AA N OW T EH K 5.13980492700294e-10
Input line 4: C Y B E R B U L L Y
(763870 states / 764362 arcs reduce-> 1141/1633)
(11458 states / 95768 arcs reduce-> 9865/85322)
S IH B ER B AH L IY 2.68538805084496e-11
S AY B ER B AH L IY 2.36700475649795e-11
S IH B ER B AH L IY 1.87574511377516e-11
S AY B ER B AH L IY 1.42886201403771e-11
S IH B ER B UH L IY 1.16880720600129e-11
Derivations found for all 4 inputs
Viterbi (best path) product of probs=e^-98.4254218705567, probability=2^-141.998 per-input-symbol-perplexity(N=38)=2^3.73679 per-line-perplexity(N=4)=2^35.4995

9) eword-epron.wfst vs. epron-espell.wfst

Words: HAMBURGER, HAMILTON, RACER, BASKETBALL

cat pron3_9.txt | ./carmel -sribIEQk 5 eword-epron.wfst
Input line 1: N ER B ER G R IH NG
(28 states / 33 arcs reduce-> 21/25)
NURRE BURGE WRING 1
NURRE BURGE RINGE 1
NURRE BURG WRING 1
NURRE BURGE RING 1
NURRE BURG RINGE 1
Input line 2: R EY IH K OW N AH N
(0 states / 0 arcs)
Empty or invalid result of composition with transducer "eword-epron.wfst".
0
0
0
0
0
Input line 3: N AA N OW T EH K
(44 states / 60 arcs reduce-> 42/58)

NAH NO TEC 1
NAH NO TEK 1
NAH NOTE EK 1
NAH NO TECK 1
NAH NOTE ECK 1
Input line 4: S AY B ER B UH L IY
(28 states / 33 arcs reduce-> 25/30)
TSAI BURR BULLY 1
TSAI BURR BULL EE 1
TSAI BIR BULLY 1
SY BURR BULLY 1
TSAI BIR BULL EE 1
No derivations found for 1 of 4 inputs
Viterbi (best path) product of probs=1, probability=2^0 per-input-symbol-perplexity(N=31)=2^-0 per-line-perplexity(N=3)=2^-0, excluding 1 0 probabilities (i.e. real ppx is infinite).

cat pron3_9.txt | ./carmel -slibOEQk 5 epron-espell.wfst
Input line 1: N ER B ER G R IH NG
(8190 states / 9022 arcs reduce-> 2367/3199)
N E R B E R G R I N G 0.709050758595687
N E R B E R G R I N G 0.684675259702072
N E R B E R G R I N 0.622654726284133
N E R B E R G R I N G 0.621411287324863
N E R B U R G R I N G 0.618313604539151
Input line 2: R EY IH K OW N AH N
(10936 states / 12124 arcs reduce-> 3274/4462)
R A Y E K O N A N 0.818813091681791
R E C O N A N 0.501248116193556
R E C O N A N 0.482085179570832
R E C O N A N 0.442377185183973
R E C O N A N 0.425464910226424
Input line 3: N AA N OW T EH K
(8314 states / 9109 arcs reduce-> 2189/2984)
K N O N O T E C H 0.704107201797722
N A N O T E C H 0.529044662277
N A N O T E C H 0.494777092178086
N O N O T E C H 0.441053174811299
N A H N O T E C H 0.427169365599
Input line 4: S AY B ER B UH L IY
(8044 states / 8839 arcs reduce-> 2202/2997)
C Y B U R B U L L Y 0.868551
S I B R E B U L L Y 0.8590536388
C Y B E R B U L L Y 0.7696113856
S I B E U R B U L L Y 0.548428406524816
C Y B U R B U L L Y 0.524308000542686
Derivations found for all 4 inputs
Viterbi (best path) product of probs=0.355055475789961, probability=2^-1.49388 per-input-symbol-perplexity(N=31)=2^0.0481898 per-line-perplexity(N=4)=2^0.373471

According to the results above, we could find the output is better on using spelling based pronunciation, if our input is phoneme sequences of new words. Because, for word based pronunciation, all outputs will combine existing words which are the most similar pronunciation with pronunciation of each part of the new word. Whereas, the spell based on pronunciation will spell every possible letter according to the pronunciation.

10) This command is to find words that have the same pronunciation as the word "BEAR".

```
echo 'BEAR' | ./carmel -sliOEQk 10 eword-epron.wfst epron-eword.wfst eword.wfsa
Input line 1: BEAR
        (209214 states / 209213 arcs reduce-> 7/6)
        (27 states / 38 arcs reduce-> 11/14)
        (11 states / 14 arcs)
BEAR 0.00010622814
BARE 1.4904459e-05
BAER 2.167921e-06
BEHR 1.3549508e-06
BAHR 5.4198027e-07
0
0
0
0
0
Derivations found for all 1 inputs
Viterbi (best path) product of probs=0.00010622814, probability=2^-13.2005 per-input-symbol-
perplexity(N=1)=2^13.2005 per-line-perplexity(N=1)=2^13.2005
```

11) Observations and Insights

→ We observed several points below:

1) **Eword-epron.wfst works well.**
2) **Espell-epron provides several candidates but there are lots of nonsense outputs.**
3) **After 2), with epron.wfsa, it provides reasonable and viable results. Epron.wfsa proves each results by its weights.**
4) **Epron to espell provides reasonable results and could find the desired results.**
5) **Epron-espell-eword-eword finds correct answers with the most highest possibilities**
6) **However, when we try epron-eword-eword, its results are much worse than 5).**
7) **Eword-epron directly does not show any results. This is because sample words are not trained before, and there are no transitions for those new words in eword-epron.wfst. Thus, a new word cannot be recognized by eword-epron.wfst.**
8) **However, espell-epron-epron provides viable results. We could pronounce it based on each letter, hence whatever the input spelling is we could get reasonable results.**
9) **So, now we compare epron-eword and epron-espell. The results support the previous findings that the performance of epron-espell is better than epron-eword.**
10) **By using pipeline with carmel, we can find words with the same pronunciation.**

As such, we could say that when we try to find English words from pronunciation, searching a word as an entity from pronunciation is not a good strategy unless if a word is in a trained data.

4. Decoding English Words from Japanese Katakana

2) LAMP

echo 'L AE M P' | ./carmel -sliOEQk 5 epron-jpron.wfst
Input line 1: L AE M P
(43 states / 64 arcs)
R A M P 0.257691717613257
R U A M P 0.149297471500533
R A M U P 0.0830020482599941
R A M P U 0.065514686817513
R A M PP U 0.0524119905487651
Derivations found for all 1 inputs
Viterbi (best path) product of probs=0.257691717613257, probability=2^-1.95628 per-input-symbol-perplexity(N=4)=2^0.48907 per-line-perplexity(N=1)=2^1.95628

3) Japanese syllable structure and the results of the experiment

→ It does not make sense. According to the Japanese syllable structure, RAMPU or RAMPPU must have been at the top. Japanese syllable cannot end with a consonant.

4) How to improve it more like Japanese

→ At the end of every word's syllable, there must be a Japanese vowels such as I, A, U, O except after N or M. Thus, we need to make sure each syllable has its matching vowels if there are consecutive consonants in the English pronunciation.

5) jprons.txt: we can guess, but it is somehow difficult to guess its english words.

cat jprons.txt | ./carmel -sribIEQk 5 epron.wfsa epron-jpron.wfst > **part4-5.txt**
Input line 1: H I R A R I K U R I N T O N
(281 states / 424 arcs reduce-> 178/321)
(3143 states / 20141 arcs reduce-> 2877/17206)
Input line 2: D O N A R U D O T O R A N P U
(357 states / 534 arcs reduce-> 228/405)
(3476 states / 22083 arcs reduce-> 3151/18968)
Input line 3: B I D E O T E E P U
(231 states / 346 arcs reduce-> 143/258)
(2288 states / 11850 arcs reduce-> 2012/9708)
Input line 4: H O M A A SH I N P U S O N
(287 states / 425 arcs reduce-> 174/312)
(2938 states / 16618 arcs reduce-> 2677/13892)
Input line 5: R A PP U T O PP U
(183 states / 281 arcs reduce-> 121/219)
(1578 states / 8676 arcs reduce-> 1354/7262)
Input line 6: SH E E B I N G U K U R I I M U
(315 states / 491 arcs reduce-> 219/395)
(3713 states / 22096 arcs reduce-> 3393/18685)
Input line 7: CH A I R U D O SH I I T O
(278 states / 426 arcs reduce-> 188/336)
(3531 states / 21544 arcs reduce-> 3228/18344)

Input line 8: SH I I T O B E R U T O
(249 states / 382 arcs reduce-> 173/306)
(2863 states / 17643 arcs reduce-> 2535/15009)
Input line 9: SH I N G U R U R U U M U
(272 states / 437 arcs reduce-> 209/374)
(4192 states / 29573 arcs reduce-> 3864/25738)
Input line 10: G A A R U H U R E N D O
(300 states / 454 arcs reduce-> 199/353)
(3814 states / 24688 arcs reduce-> 3561/21255)
Input line 11: T O R A B E R A A Z U TCH E KK U
(372 states / 561 arcs reduce-> 232/421)
(3999 states / 28010 arcs reduce-> 3707/24303)
Input line 12: B E B I I SH I TT A A
(222 states / 337 arcs reduce-> 143/258)
(2469 states / 10711 arcs reduce-> 1879/8404)
Input line 13: S U K O TT O R A N D O
(249 states / 369 arcs reduce-> 154/274)
(2104 states / 11771 arcs reduce-> 1893/9861)
Input line 14: B A I A R I N K O N TCH E R U T O
(366 states / 549 arcs reduce-> 232/415)
(4333 states / 29320 arcs reduce-> 3948/25265)
Input line 15: A PP U R U M A KK U B U KK U P U R O
(397 states / 618 arcs reduce-> 269/490)
(3725 states / 23708 arcs reduce-> 3316/20460)
Input line 16: K O N P I U U T A S A I E N S U
(392 states / 598 arcs reduce-> 253/459)
(5562 states / 39252 arcs reduce-> 5146/34037)
Input line 17: H I J I K A R U T O R E E N I N G U
(400 states / 610 arcs reduce-> 270/480)
(4768 states / 31057 arcs reduce-> 4349/26711)
Input line 18: H I J I K A R U E K U S A S A I S U
(439 states / 676 arcs reduce-> 294/531)
(5819 states / 41154 arcs reduce-> 5292/35756)
Input line 19: A I S U K U R I I M U
(267 states / 422 arcs reduce-> 189/344)
(3397 states / 22032 arcs reduce-> 3108/18900)
Input line 20: H O TT O M I R U K U
(211 states / 322 arcs reduce-> 140/251)
(1916 states / 11777 arcs reduce-> 1619/10046)
Input line 21: T O R I P U R U R U U M U
(310 states / 493 arcs reduce-> 228/411)
(4543 states / 32541 arcs reduce-> 4196/28349)
Input line 22: K U R A U N P U R A Z A H O T E R U
(448 states / 682 arcs reduce-> 287/521)
(5382 states / 36444 arcs reduce-> 4809/31381)
Input line 23: H E E S U B U KK U R I S A A TCH I S A I E N T I S U T O
(636 states / 972 arcs reduce-> 417/753)
(7909 states / 50903 arcs reduce-> 7341/43647)
Input line 24: W O R U H U G A N G U M O TS U A R U T O

(473 states / 726 arcs reduce-> 324/577)
(5765 states / 36414 arcs reduce-> 5281/31303)
Derivations found for all 24 inputs
Viterbi (best path) product of probs=e^-959.974412832776, probability=2^-1384.95 per-input-symbol-perplexity(N=342)=2^4.04956 per-line-perplexity(N=24)=2^57.7063

6) By assembling eword.wfsa, eword-epron.wfst, and epron-jpron.wfst: it does make sense. It provides viable suggestions based on the input. This is because of the language recognition system. There are two ways to recognize words: by vocal or by letters. Those are cognitively different systems. In this experiment, we are not listening to the pronunciation but looking at it. Of course, looking at it and recognizing it has more benefits in this kind of experiments. We can recognize words by letters no matter how pronunciations are different.

cat jprons.txt | ./carmel -sribIEQk 5 eword.wfsa eword-epron.wfst epron-jpron.wfst > **part4-6.txt**
Input line 1: H I R A R I K U R I N T O N
(281 states / 424 arcs reduce-> 178/321)
(24777 states / 33877 arcs reduce-> 10855/15014)
(9888 states / 13080 arcs reduce-> 8852/12044)
Input line 2: D O N A R U D O T O R A N P U
(357 states / 534 arcs reduce-> 228/405)
(30480 states / 41482 arcs reduce-> 13914/19218)
(12721 states / 16832 arcs reduce-> 11407/15515)
Input line 3: B I D E O T E E P U
(231 states / 346 arcs reduce-> 143/258)
(13238 states / 17542 arcs reduce-> 5295/7400)
(4855 states / 6520 arcs reduce-> 4539/6204)
Input line 4: H O M A A SH I N P U S O N
(287 states / 425 arcs reduce-> 174/312)
(16106 states / 21494 arcs reduce-> 6641/9068)
(6065 states / 7916 arcs reduce-> 5355/7205)
Input line 5: R A PP U T O PP U
(183 states / 281 arcs reduce-> 121/219)
(10974 states / 14684 arcs reduce-> 4834/6660)
(4462 states / 5916 arcs reduce-> 4070/5524)
Input line 6: SH E E B I N G U K U R I I M U
(315 states / 491 arcs reduce-> 219/395)
(26738 states / 36331 arcs reduce-> 11314/15988)
(10193 states / 13746 arcs reduce-> 9224/12777)
Input line 7: CH A I R U D O SH I I T O
(278 states / 426 arcs reduce-> 188/336)
(20240 states / 27677 arcs reduce-> 9140/12695)
(8377 states / 11169 arcs reduce-> 7518/10310)
Input line 8: SH I I T O B E R U T O
(249 states / 382 arcs reduce-> 173/306)
(24591 states / 33552 arcs reduce-> 10942/15287)
(9950 states / 13303 arcs reduce-> 8932/12284)
Input line 9: SH I N G U R U R U U M U
(272 states / 437 arcs reduce-> 209/374)
(58550 states / 79016 arcs reduce-> 25107/35126)

(22710 states / 30332 arcs reduce-> 20085/27700)
Input line 10: G A A R U H U R E N D O
(300 states / 454 arcs reduce-> 199/353)
(24754 states / 33730 arcs reduce-> 10860/15133)
(9995 states / 13403 arcs reduce-> 9089/12495)
Input line 11: T O R A B E R A A Z U TCH E KK U
(372 states / 561 arcs reduce-> 232/421)
(32926 states / 44219 arcs reduce-> 14621/20132)
(13503 states / 17896 arcs reduce-> 12311/16701)
Input line 12: B E B I I SH I TT A A
(222 states / 337 arcs reduce-> 143/258)
(9370 states / 12654 arcs reduce-> 3847/5494)
(3401 states / 4602 arcs reduce-> 3036/4237)
Input line 13: S U K O TT O R A N D O
(249 states / 369 arcs reduce-> 154/274)
(13887 states / 18978 arcs reduce-> 6342/8852)
(5842 states / 7852 arcs reduce-> 5370/7379)
Input line 14: B A I A R I N K O N TCH E R U T O
(366 states / 549 arcs reduce-> 232/415)
(33331 states / 45262 arcs reduce-> 14984/20764)
(13622 states / 18040 arcs reduce-> 12212/16630)
Input line 15: A PP U R U M A KK U B U KK U P U R O
(397 states / 618 arcs reduce-> 269/490)
(29678 states / 40326 arcs reduce-> 12200/17475)
(11092 states / 15259 arcs reduce-> 10216/14383)
Input line 16: K O N P I U U T A S A I E N S U
(392 states / 598 arcs reduce-> 253/459)
(56101 states / 75148 arcs reduce-> 25242/34413)
(23141 states / 30211 arcs reduce-> 20460/27527)
Input line 17: H I J I K A R U T O R E E N I N G U
(400 states / 610 arcs reduce-> 270/480)
(53805 states / 73129 arcs reduce-> 25563/34855)
(23531 states / 30791 arcs reduce-> 21215/28465)
Input line 18: H I J I K A R U E K U S A S A I S U
(439 states / 676 arcs reduce-> 294/531)
(67918 states / 91680 arcs reduce-> 29860/41310)
(27247 states / 36084 arcs reduce-> 24594/33423)
Input line 19: A I S U K U R I I M U
(267 states / 422 arcs reduce-> 189/344)
(31003 states / 41477 arcs reduce-> 13108/18185)
(11912 states / 15793 arcs reduce-> 10663/14542)
Input line 20: H O TT O M I R U K U
(211 states / 322 arcs reduce-> 140/251)
(14206 states / 19407 arcs reduce-> 5780/8336)
(5130 states / 7036 arcs reduce-> 4575/6479)
Input line 21: T O R I P U R U R U U M U
(310 states / 493 arcs reduce-> 228/411)
(63623 states / 85372 arcs reduce-> 27347/38035)
(24845 states / 33031 arcs reduce-> 22049/30228)

Input line 22: K U R A U N P U R A Z A H O T E R U
(448 states / 682 arcs reduce-> 287/521)
(48769 states / 66276 arcs reduce-> 21467/30141)
(19633 states / 26473 arcs reduce-> 17862/24702)
Input line 23: H E E S U B U KK U R I S A A TCH I S A I E N T I S U T O
(636 states / 972 arcs reduce-> 417/753)
(54588 states / 73286 arcs reduce-> 23777/32952)
(21695 states / 28790 arcs reduce-> 19556/26650)
Input line 24: W O R U H U G A N G U M O TS U A R U T O
(473 states / 726 arcs reduce-> 324/577)
(53468 states / 73203 arcs reduce-> 23847/33185)
(21552 states / 28587 arcs reduce-> 18981/26006)
Derivations found for all 24 inputs
Viterbi (best path) product of probs=e^-741.961427596585, probability=2^-1070.42 per-input-symbol-perplexity(N=342)=2^3.12989 per-line-perplexity(N=24)=2^44.601

7) Some desired outputs were not ranked top. Why? How would you improve the results?

APPLE MAKE BOOK PRO 1.84704262710222e-22
APPLE MAC BOOK PRO 1.79096349941989e-22
OPERA MAKE BOOK PRO 9.09864953151017e-23
OPERA MAC BOOK PRO 8.82240017953144e-23
APPLE MAKE BOOK PLOUGH 2.05215303768608e-23

→ As you can see above, there is a problem of "MAKE" and "MAC". "MAKE" has higher possibility than "MAC". Specifically, After the first syllable, "MA", each has consonant, "K" or "C". "K" has more chances than "C", and once "K"7) Some is chosen, it tends to have "E" to make "MAKE", but "C" itself has, both empirically and statistically, less chance to have a vowel according to the trained data set. Also, when we consider APPLE as a noun subject, it could be natural there will be a verb than a noun. These two things made the desired output, "APPLE MAC BOOK PRO", was not top ranked. Even worse, actually, it was not exactly the desired output. It should have been "APPLE MACBOOK PRO", but it happened because Japanese needs to make each syllable with a vowel, so, it separated MAC and BOOK. In order to improve the results, it is better to study compounding words and its syllable combination rule. Then, it would be applied to the dataset with appropriate possibilities.

8) part4-8_katakana.txt → compounding words

cat part4-8_katakana.txt | ./carmel -sribIEQk 5 eword.wfsa eword-epron.wfst epron-jpron.wfst
Input line 1: S A N K U S U G I B I N G U D E
(321 states / 489 arcs reduce-> 212/380)
(15802 states / 21267 arcs reduce-> 6468/9162)
(5759 states / 7744 arcs reduce-> 5130/7115)
THANKSGIVING DAY 2.6527116015287e-12
THANKSGIVING DES 8.09161185511585e-13
THANKSGIVING DU 1.73298109615673e-13
THANKSGIVING GOODS 4.58368374418332e-14
THANKSGIVING DES 4.24930735895522e-14
Input line 2: O R A N D A
(155 states / 225 arcs reduce-> 87/157)
(6207 states / 8601 arcs reduce-> 2747/3938)
(2518 states / 3480 arcs reduce-> 2374/3335)

ALL UNDER 2.62184630865706e-08
AROUND 8.50283636150492e-10
OIL UNDER 5.80564365954105e-10
ORE UNDER 4.25619161749324e-10
TRENDS 1.98589095220964e-10
Input line 3: R E A R U M A D O R I D O
(314 states / 474 arcs reduce-> 202/362)
(29908 states / 40656 arcs reduce-> 12222/17310)
(11064 states / 14986 arcs reduce-> 9984/13903)
REAL MADRID 7.35699865865463e-14
RARE MADRID 3.53616754613867e-14
RAIL MADRID 1.28699074287164e-14
LOWELL MADRID 1.21189562764494e-14
RARELY MADRID 1.03279692232131e-14
Input line 4: K I M U K A D A S H I A N
(276 states / 412 arcs reduce-> 169/305)
(11758 states / 15983 arcs reduce-> 4592/6456)
(4181 states / 5634 arcs reduce-> 3867/5320)
COME CADAM WHEN 2.00249568092255e-21
COME CADAM FEARS 9.29373788311536e-22
KIM CADAM WHEN 8.04909615286316e-22
COME CADAM WHEN 7.57167997547183e-22
COMES CADAM WHEN 4.39362998458009e-22
Input line 5: Y O G U R U T O S U M U J I
(322 states / 499 arcs reduce-> 222/399)
(40927 states / 55662 arcs reduce-> 18961/26018)
(17254 states / 22604 arcs reduce-> 15251/20597)
YOGURT SOME DES 8.20194231618379e-21
YOGURT SOME SEE 4.5845031065685e-21
YOGURT SOME AGE 3.51174599329582e-21
YOGI ROUTES MOODY 3.4022990101488e-21
YOGURT OS MOODY 1.79999366794825e-21
Derivations found for all 5 inputs
Viterbi (best path) product of probs=e^-168.262588641506, probability=2^-242.752 per-input-symbol-perplexity(N=62)=2^3.91535 per-line-perplexity(N=5)=2^48.5503

9) Input jpron.txt right to left
epron-jpron.wfst generate a file
Then input this file, left to right epron-espell.wfst espell-eword.wfst eword.wfsa

10)We could decode the jprons to english words by transferring jpron to epron, then epron to espell and check the correction of espell. So for each wrong pronunciation, we could also find a english spelling even if it is not a english word, but if we see that incorrect word, we are able to recognize what exactly people want to express according to their pronunciation if those written languages are originated from Roman alphabets.