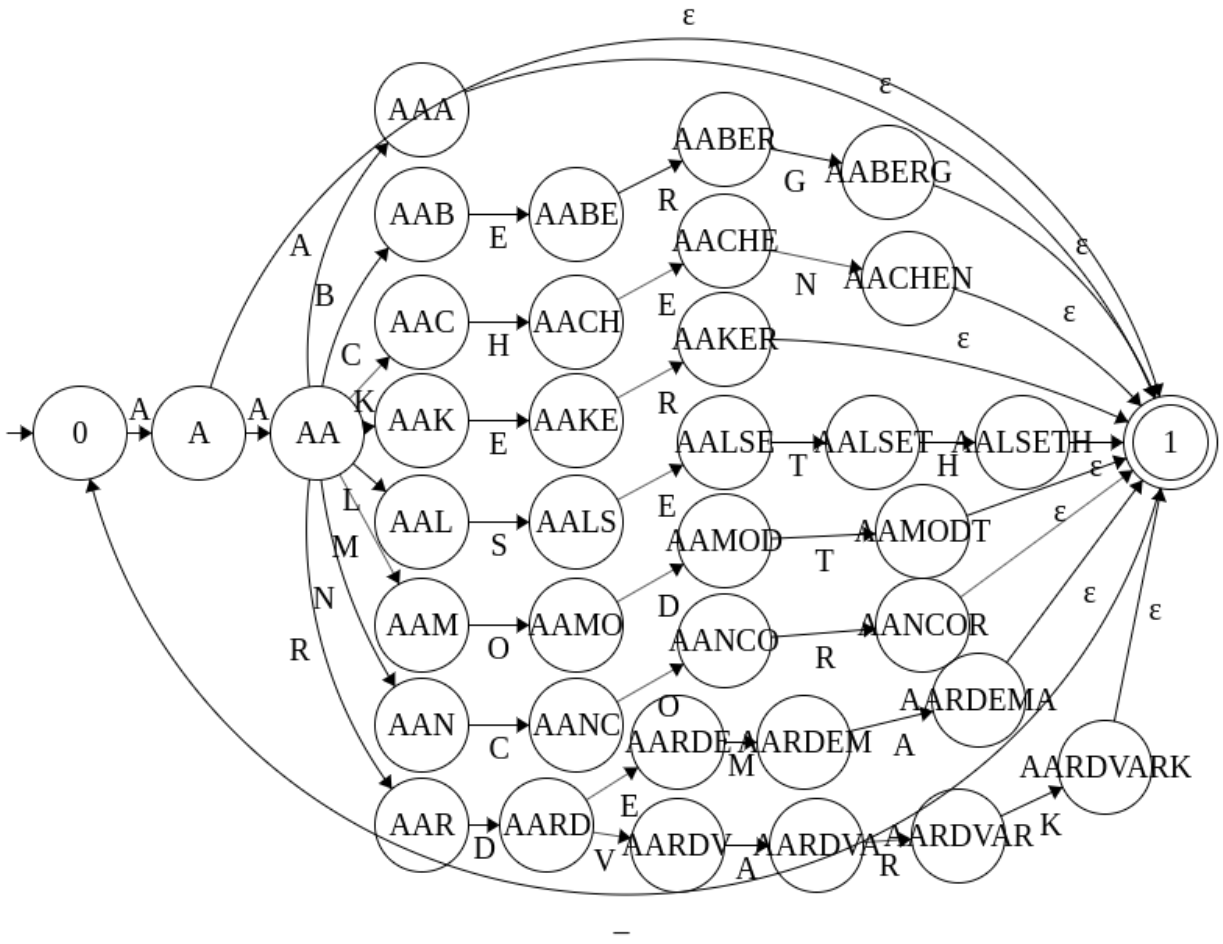
NLP HW1

1. **Finite-state acceptors**

1-1.



1-2.

(a)

Number of states in result: 256331

Number of arcs in result: 361732

Number of paths in result (valid for acyclic only; a cycle means infinitely many): 105402.000000005

There are 361732 transitions, and 256331 states.

(b)

Strings:

Input line 1: L I S T _ T H E _ F L I G H T S _ F R O M _ B A L T I M O R E _ T O _ S E A T T L E _ T H A T _ S T O P _ I N _ M I N N E A P O L I S

(102 states / 101 arcs reduce-> 79/78)

L I S T _ T H E _ F L I G H T S _ F R O M _ B A L T I M O R E _ T O _ S E A T T L E _ T H A T _ S T O P _ I N _ M I N N E A P O L I S

Input line 2: D O E S _ T H I S _ F L I G H T _ S E R V E _ D I N N E R
(48 states / 47 arcs reduce-> 35/34)
D O E S _ T H I S _ F L I G H T _ S E R V E _ D I N N E R

Input line 3: I _ N E E D _ A _ F L I G H T _ T O _ S E A T T L E _ L E A V I N G _ F R O M _ B A L T I M O R E _ M A K I N G _ A _ S T O P _ I N _ M I N N E A P O L I S
(119 states / 118 arcs reduce-> 93/92)
I _ N E E D _ A _ F L I G H T _ T O _ S E A T T L E _ L E A V I N G _ F R O M _ B A L T I M O R E _ M A K I N G _ A _ S T O P _ I N _ M I N N E A P O L I S

Input line 4: I _ N E E D _ T O _ H A V E _ D I N N E R _ S E R V E D
(48 states / 47 arcs reduce-> 35/34)
I _ N E E D _ T O _ H A V E _ D I N N E R _ S E R V E D

Input line 5: I _ H A V E _ T W O _ F R I E N D S _ T H A T _ W O U L D _ L I K E _ T O _ V I S I T _ M E _ O N _ W E D N E S D A Y _ H E R E _ I N _ W A S H I N G T O N _ D _ C
(128 states / 127 arcs reduce-> 100/99)
I _ H A V E _ T W O _ F R I E N D S _ T H A T _ W O U L D _ L I K E _ T O _ V I S I T _ M E _ O N _ W E D N E S D A Y _ H E R E _ I N _ W A S H I N G T O N _ D _ C

strings.bad:
Input line 1: I _ W U N T _ T O _ L E E V E _ M O N D A Y _ M O R N I N G
(0 states / 0 arcs)
Empty or invalid result of composition with transducer "english.fsa".

Input line 2: N O W _ I _ N E A D _ A _ F L I G H T _ O N _ T O O S D A Y _ F R O M _ P H E E N I X _ T O _ D E T R O I T
(0 states / 0 arcs)
Empty or invalid result of composition with transducer "english.fsa".

Input line 3: W H I C H _ O N E S _ L E E V E _ I N _ T H E _ M O R N I N G
(0 states / 0 arcs)
Empty or invalid result of composition with transducer "english.fsa".

Input line 4: W H I C H _ O N E S _ A R R I V E _ E R L Y _ I N _ T H E _ D A Y

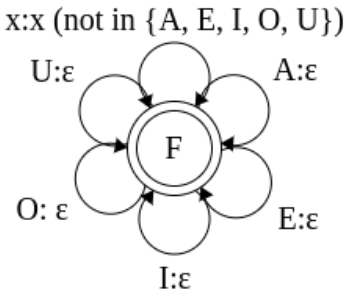
(0 states / 0 arcs)
Empty or invalid result of composition with transducer "english.fsa".

Input line 5: I _ N E A D _ A _ F L I G H T _ F R O M _ P H E E N I X _ T O _ D E T R O I T _ L E E V I N G _ M O N D A Y _ E E V E N I N G
(0 states / 0 arcs)
Empty or invalid result of composition with transducer "english.fsa".

2. Finite-state transducers

2-3.

(a)



(b)
L S T _ T H _ F L G H T S _ F R M _ B L T M R _ T _ S T T L _ T H T _ S T P _ N _ M N N P L S
D S _ T H S _ F L G H T _ S R V _ D N N R
_ N D _ _ F L G H T _ T _ S T T L _ L V N G _ F R M _ B L T M R _ M K N G _ _ S T P _ N _ M N N P L S
_ N D _ T _ H V _ D N N R _ S R V D
_ H V _ T W _ F R N D S _ T H T _ W L D _ L K _ T _ V S T _ M _ N _ W D N S D Y _ H R _ N _ W S H N G T N _ D _ C

(c)
B L D N G
B L D N G U
B L D N U G
B L D N G U U
B L D N U G U
B L D N G U U U
B L D N U G U U
B L D N G U U U U
B L D N U G U U U

B L D N G U U U U U

2-4.

(a)

L S T _ T H _ F L G H T S _ F R M _ B L T M R _ T _ S T T L _ T H T _ S T P _ N _ M N N P L S

D S _ T H S _ F L G H T _ S R V _ D N N R

_ N D _ _ F L G H T _ T _ S T T L _ L V N G _ F R M _ B L T M R _ M K N G _ _ S T P _ N _ M N N P L S

_ N D _ T _ H V _ D N N R _ S R V D

_ H V _ T W _ F R N D S _ T H T _ W L D _ L K _ T _ V S T _ M _ N _ W D N S D Y _ H R _ N _ W S H N G T N _ D _ C

(b)

The accuracy is 1.2865497076%

(c)

Because automaton could generate an infinite number of words by randomly inserting each vowel, and the majority of words are meaningless which looks randomly organized. And automaton will select the first word which is output itself (without vowel letters). Therefore, it is nearly impossible to get the correct word for each non-vowel word (except words like “S H Y”,etc.). So the accuracy of vowel restorer is very low.

3. Combining FSAs and FSTs

3-5.

(a)

First of all, we generated a FSA for file *vocab*. Then, based on that FSA, we did some modifications in python file. If the input is one of the vowels, the output will be modified to *e*, instead of the input itself. Finally, we combine the FSA with FST.

We could also achieve combination by using command:

cat strings | carmel -slibOEWk 1 english.fsa remove-vowels.fst

(b)

The accuracy is 33.0994152047%

make.py is our python file for implementing our combined vowel restorer.

vowels_remove.fst is our combined FST. *strings.restored.mod1* is the restored samples of the implementation.

(c)

The accuracy is still low and the results are not satisfactory. People are based on grammar, context meaning and the correctness of the word to restore the vowels for

English words. However, in this case, our combined FSA and FST automaton only based on word correctness. It will lead to random guess to restore the vowels without thinking about the grammatical problem and sentence meaning, when several words are same after running *vowels.remove*.

3-6.

(a)

First of all, we could generate *english.fsa* by using test file (*strings*) instead of the *vocab* file which includes much more words. Our restorer will be more likely to recover correctly since the fewer number of words could be derived in smaller *english.fsa*.

For the further improve, we add weights in each path, and automaton will know the most possible vowel letter which will follow by the state (identified string). Hence automaton is able to find the most likely word based on the frequency of words in *strings* which will improve the accuracy a lot.

(b)

Question6.py is our python file for implementing our improved vowel restorer.

Vowels_remove.wfst is our improved FST with adding probabilities in it.

strings.restored.mod2 is the restored samples of the implementation.

(c)

The accuracy, after implementing two ideas above, improves to 91.35%.