ECSE-526 Assignment # 2

Muhammad Amir Hamza
McGill ID : 261210478

October 8, 2024

1 Abstract

The purpose of this assignment was to explore the applications of Markov and Hidden Markov Models in natural language processing. The assignment consisted of two tasks: sequence of states generation from second order Markov Model and identifying the sequence of hidden states using first order Hidden Markov Model. The first task was applied to generate sentences. In second task, the application was to correct the given sentences. Both tasks were coded in Python, and expected results were successfully achieved.

2 Introduction

Markov Model is probabilistic model in which the next state only depends upon the fixed number of previous states. This fixed number of previous states denotes the order of that Markov Model. For example, first order Markov Model will only depend upon most recent last state. In the given assignment, the task was to simulate second order Markov Model to get a sequence of output states, where each output state denotes a word from English vocabulary. Resulting in a generation of full sentence using this probability distribution given in *trigram* dataset.

In the second part of the assignment, the task was to correct the given wrong sentences by considering it as the observation of a first order Hidden Markov Model and then finding correct sentence by finding the underlying hidden sequence of states. For the emission probability of observation and the hidden state, the task considered the Poisson distribution as,

$$P(O_t = u|X_t = v) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{1}$$

Where k is the Levenshtein distance between the string u and v. To calculate the Levenshtein distance, the author is thankful to the Maintainers of **python-Levenshtein project** (https://pypi.org/project/python-Levenshtein/). This saved a lot of time and help to focus on the given task.

3 Outputs of Part # 1

The outputs of part # 1, with the tagging of the distribution is given below, it should be noted that back up was implemented in the algorithm but there are a few cases in which we actually require backup.

```
Sentences generated by 2nd Order Markov
↪  model

1    <s>(predefined)  I(bigram)
↪  have(trigram)  no(trigram)
↪  ambition(trigram)  ,(trigram)
↪  I(trigram)  suppose(trigram)
↪  ,(trigram)  is(trigram)  not(trigram)
↪  to(trigram)  have(trigram)  a(trigram)
↪  great(trigram)  while(trigram)
↪  ;(trigram)  and(trigram)  I(trigram)
↪  will(trigram)  not(trigram)
↪  spare(trigram)  him(trigram)
↪  .(trigram)  </s>(trigram)


2    <s>(predefined)  "(bigram)
↪  Dear(trigram)  ,(trigram)
↪  dear(trigram)  Norland(trigram)
↪  ,"(trigram)  said(trigram)
↪  Anne(trigram)  ,(trigram)  by(trigram)
↪  way(trigram)  of(trigram)
↪  love(trigram)  .(trigram)
↪  </s>(trigram)
```

```
3    <s>(predefined)  If(bigram)
↪  a(trigram)  man(trigram)  ,(trigram)
↪  that(trigram)  he(trigram)
↪  is(trigram)  ,(trigram)  I(trigram)
↪  believe(trigram)  ,(trigram)
↪  had(trigram)  she(trigram)
↪  so(trigram)  much(trigram)
↪  of(trigram)  her(trigram)  as(trigram)
↪  deeply(trigram)  and(trigram)
↪  as(trigram)  if(trigram)  he(trigram)
↪  were(trigram)  ever(trigram)
↪  got(trigram)  over(trigram)
↪  ,(trigram)  and(trigram)
↪  admire(trigram)  him(trigram)
↪  at(trigram)  least(trigram)
↪  by(trigram)  the(trigram)
↪  entrance(trigram)  of(trigram)
↪  her(trigram)  life(trigram)
↪  ,(trigram)  a(trigram)  bush(trigram)
↪  of(trigram)  low(trigram)
↪  company(trigram)  ,(trigram)
↪  and(trigram)  I(trigram)
↪  must(trigram)  have(trigram)
↪  owed(trigram)  a(trigram)
↪  wife(trigram)  ,(trigram)
↪  who(trigram)  called(trigram)
↪  on(trigram)  her(trigram)
↪  side(trigram)  ,(trigram)
↪  Elinor(trigram)  ,"(trigram)
↪  she(trigram)  cried(trigram)
↪  ,(trigram)  "(trigram)  I(trigram)
↪  believe(trigram)  ,(trigram)
↪  with(trigram)  some(trigram)
↪  of(trigram)  her(trigram)
↪  mother(trigram)  '(trigram)
↪  s(trigram)  eyes(trigram)
↪  expressed(trigram)  the(trigram)
↪  astonishment(trigram)  which(trigram)
↪  her(trigram)  sister(trigram)
↪  .(trigram)  </s>(trigram)
```

```
4    <s>(predefined)  Such(bigram)
↪  talent(trigram)  as(trigram)
↪  hers(trigram)  must(trigram)
↪  not(trigram)  be(trigram)
↪  sorry(trigram)  to(trigram)
↪  have(trigram)  known(trigram)
↪  you(trigram)  long(trigram)
↪  to(trigram)  attempt(trigram)
↪  the(trigram)  walk(trigram)
↪  must(trigram)  arise(trigram)
↪  from(trigram)  the(trigram)
↪  observation(trigram)  of(trigram)
↪  the(trigram)  first(trigram)
↪  week(trigram)  .(trigram)
↪  </s>(trigram)
```

```
5    <s>(predefined)  I(bigram)
↪  assure(trigram)  you(trigram)
↪  ,(trigram)  and(trigram)
↪  without(trigram)  being(trigram)
↪  at(trigram)  Hartfield(trigram)
↪  ,(trigram)  and(trigram)
↪  with(trigram)  a(trigram)
↪  smile(trigram)  ,(trigram)
↪  and(trigram)  she(trigram)
↪  had(trigram)  never(trigram)
↪  been(trigram)  admitted(trigram)
↪  to(trigram)  join(trigram)
↪  or(trigram)  to(trigram)  see(trigram)
↪  her(trigram)  ,(trigram)  was(trigram)
↪  peculiarly(trigram)
↪  gratifying(trigram)  .(trigram)
↪  </s>(trigram)
```

4 Outputs of Part # 2

For the given incorrect sentences, the response of our alogithm is given below:

```
Sentence \#  1
---------------
Incorrect Sentence       :  I think hat
↪  twelve thousand pounds
Corrected Sentence       :  I think at
↪  twelve thousand pounds
Max Prob                 :  -13.32422
Time taken to processe   :
↪  0.3123054504394531
```

```
Sentence \#  2
---------------
Incorrect Sentence       :  she haf heard
↪  them
Corrected Sentence       :  she had heard
↪  them
Max Prob                 :
↪  -10.498862999999998
Time taken to processe   :
↪  0.1638784408569336
```

```
Sentence \#  3
---------------
Incorrect Sentence       :  She was
↪  ulreedy quit live
Corrected Sentence       :  She was
↪  already quite like
```

```
Max Prob                  :
↪  -18.13562799566398
Time taken to processe    :
↪  0.2201242446899414



Sentence \#  4
---------------
Incorrect Sentence        :  John Knightly
↪  wasn’t hard at work
Corrected Sentence        :  John Knightley
↪  was hard at work
Max Prob                  :
↪  -20.187092250383643
Time taken to processe    :
↪  0.2668619155883789



Sentence \#  5
---------------
Incorrect Sentence        :  he said nit
↪  word by
Corrected Sentence        :  he said it
↪  would be
Max Prob                  :
↪  -17.135994995663978
Time taken to processe    :
↪  0.21351194381713867
```

5 Project Directory

In the project directory, the task # 1 was implemented in *Markov_Model_order2.py* and task # 2 was implemented in *Decoding_sequence.py*. The results of task # 1 and task # 2 are in *task1:sentences.txt* and *task2 corrected_sentences.txt*, respectively.