

Project 03 Report : Word Cross

Team Members

Ashish Goel (109753528)

Rohan Bharadwaj (109758985)

Shashank Jain (109956091)

Summary

`python wordCross.py -q1 -q2 -q3 -q4 -q5`

Question 1 accuracy: 0.311102

Q1 time used: 0.560851 secs.

Question 2 accuracy: 0.089656 (two correct letters)

Question 2 accuracy: 0.360026 (at least one correct letter)

Q2 time used: 14.686833 secs.

Question 3 accuracy: 0.143352

Q3 time used: 5.448987 secs.

Question 4 accuracy: 0.435074

Q4 time used: 5.406670 secs.

Question 5 accuracy: 0.465159

Q5 time used: 0.890978 secs.

In almost all of the equations, we aren't considering probability of factors where both the characters are known, say like, $P(a|a), P(b|')$ because we've to report the maximum value, therefore it'd be same in all configurations, thus we can successfully ignore it.

Question 1

Implementation

We concatenate the query in between “ ` “, so that we can keep track of beginning and start of the word. Then we find out the index of the missing element. We loop over all the alphabets and replace the missing letter to find out the probability and store the probabilities in a list. Finally we take the max value from this list and return corresponding alphabet.

Result

python wordCross.py -q1

Question 1 accuracy: 0.311102

Q1 time used: 0.560851 secs.

python wordCross.py -q1 --test

Question 1 accuracy: 1.000000

Q1 time used: 0.000494 secs.

Question 2

Implementation

We concatenate the query in between “ ` ”, so that we can keep track of beginning and start of the word. Then we find out the index of both the missing elements. We loop over all the alphabets for first missing letter and then loop over inside (nested loop) for the second missing letter by replacing them to find out the probability and store the probabilities in a dictionary. Finally we take the max value from this dictionary and return corresponding alphabets.

Result

python wordCross.py -q2

Question 2 accuracy: 0.089656 (two correct letters)

Question 2 accuracy: 0.360026 (at least one correct letter)

Q2 time used: 14.686833 secs.

python wordCross.py -q2 --test

Question 2 accuracy: 1.000000 (two correct letters)

Question 2 accuracy: 1.000000 (at least one correct letter)

Q2 time used: 0.008777 secs.

Question 3

Implementation

We're representing each hidden variable indepently, say like, first occurring hidden variable is represented as '1-' and so on. To gain on time benefits, we're storing temporal results of one level variable elimination. Following are the details of steps being followed: _

- 1) Convert the query into various factors involving hidden variables. Give unique names to hidden variables.
- 2) Remove a single level of hidden variable by using and saving temporal results.
- 3) Remove the rest of hidden variables, so that we're just down to two factors involving missing variables.

- 4) Now we just need to do what we did in Question1), i.e, substitute various characters to check which gives highest value and simply return it.

Result

python wordCross.py -q3

Question 3 accuracy: 0.143352

Q3 time used: 5.448987 secs.

python wordCross.py -q3 --test

Question 3 accuracy: 0.857143

Q3 time used: 0.024268 secs.

Question 4

Implementation

- 1) Final probability of missing variable can be reduced down to computing joint probability of all 4 words, which can be reduced down to product of joint probability of each single word.
- 2) Use implementation in Q3) to get the two factors involving missing variable for each of the given word.
- 3) Multiply everything from step2) and do simple character substitution to check which returns the highest value, which would be our answer.

Result

python wordCross.py -q4

Question 4 accuracy: 0.435074

Q4 time used: 5.406670 secs

python wordCross.py -q4 --test

Question 4 accuracy: 1.000000

Q4 time used: 0.033202 secs.

Question 5

Implementation

We concatenate the query in between “ ` ”, so that we can keep track of beginning and start of the word. Then we find out the index of the missing element. We loop over all the alphabets and replace the missing letter to find out the probability and store the probabilities in a list. Finally we take the max value from this list and return corresponding alphabet.

Result

python wordCross.py -q5

Question 5 accuracy: 0.465159

Q5 time used: 0.890978 secs.

python wordCross.py -q5 --test

Question 5 accuracy: 1.000000

Q5 time used: 0.000721 secs.