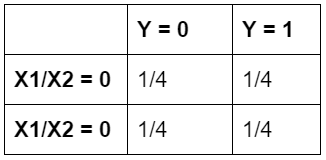
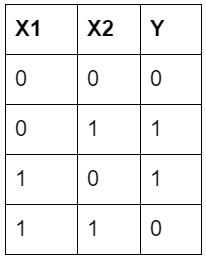
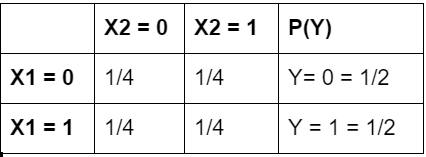
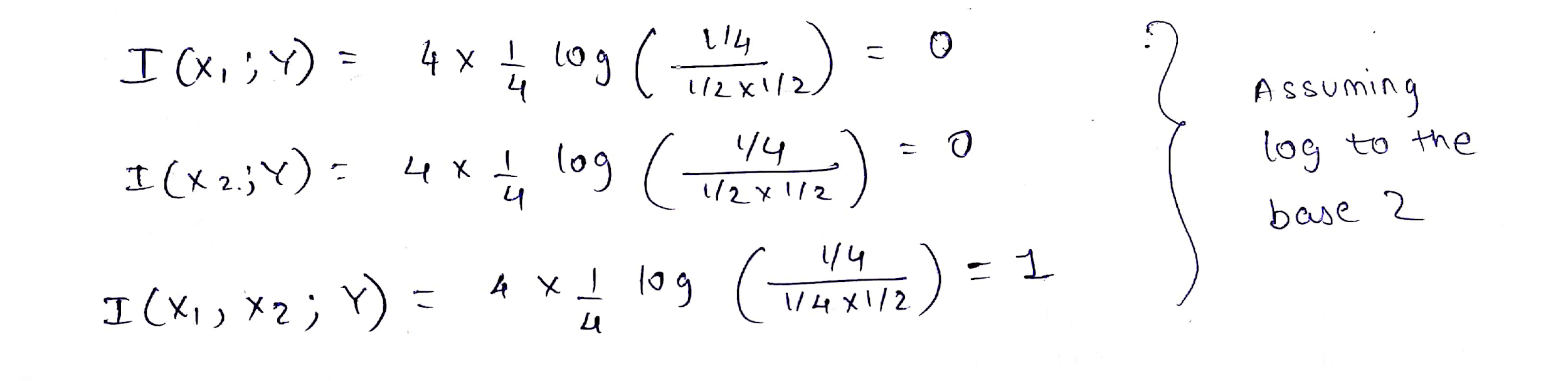
**Summer Semester 2020 SNLP Assignment 5**

**Name: Awantee Deshpande  
Id: 2581348  
Email:** [**s8awdesh@stud.uni-saarland.de**](mailto:s8awdesh@stud.uni-saarland.de)

**Name: Lakshmi Rajendra Bashyam  
Id: 2581455  
Email:** [**s8laraje@stud.uni-saarland.de**](mailto:s8awdesh@stud.uni-saarland.de)

**1 Mutual Information**

1.1 Take the following example of a simple XOR gate  
  




Thus, it is incorrect to say that if I(X1;Y) = I(X2;Y) = 0, then I(X1,X2;Y) = 0.

1.2 Mutual information and entropy

Given that the answer A is a deterministic function of the object o and the question q i.e. A = f(o,q).  
Then, since entropy of a deterministic variable is zero, we can infer that H(A|Q,O) = 0. ---(1)  
  
It is also given that object O and question Q are independent.   
Thus, H(Q|O) = H(Q) … from the definition of independent variables ---(2)  
  
We know that, I(X;Y) = H(Y) - H(Y|X)

Thus, I(O;Q,A) = H(Q,A) - H(Q,A| O) ---(3)

Using H(X,Y) = H(Y|X) + H(X),   
H(Q,A) = H(A|Q) + H(Q) ---(4)  
  
Using the chain rule for conditional entropy, H(X, Y |Z) = H(X|Z) + H(Y|X, Z),   
H(Q,A | O) = H(Q|O) + H(A|Q,O)  
Substituting the corresponding values from (1), and (2),  
H(Q,A | O) = H(Q) + 0 = H(Q) ---(5)

Plugging in (4), and (5) in (3) gives us  
I(O;Q,A) = [H(A|Q) + H(Q)] - [H(Q)]

∴ I(O;Q,A) = H(A|Q)

Hence, proved.

**2 Encoding**

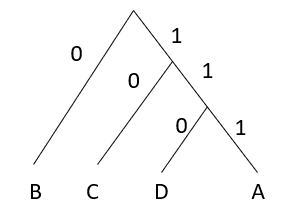
2.1 Huffman encoding

Given string: BBBDBACCDBDBACC

Character frequencies:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| frequency | 2 | 6 | 4 | 3 |
| probability | 0.1333 | 0.4 | 0.2667 | 0.2 |

The corresponding binary encoding tree is obtained by sorting the characters by their frequencies/probabilities as follows:



A = 111

B = 0

C = 10

D = 110

The encoded string is:

00011001111010110011001111010

This has a length of 29.

For each character above, the optimal code length is given by



Here, D = 2. Thus we obtain the li as follows:

lA = -log2(0.1333) = 2.9

lB = -log2(0.4) = 1.32

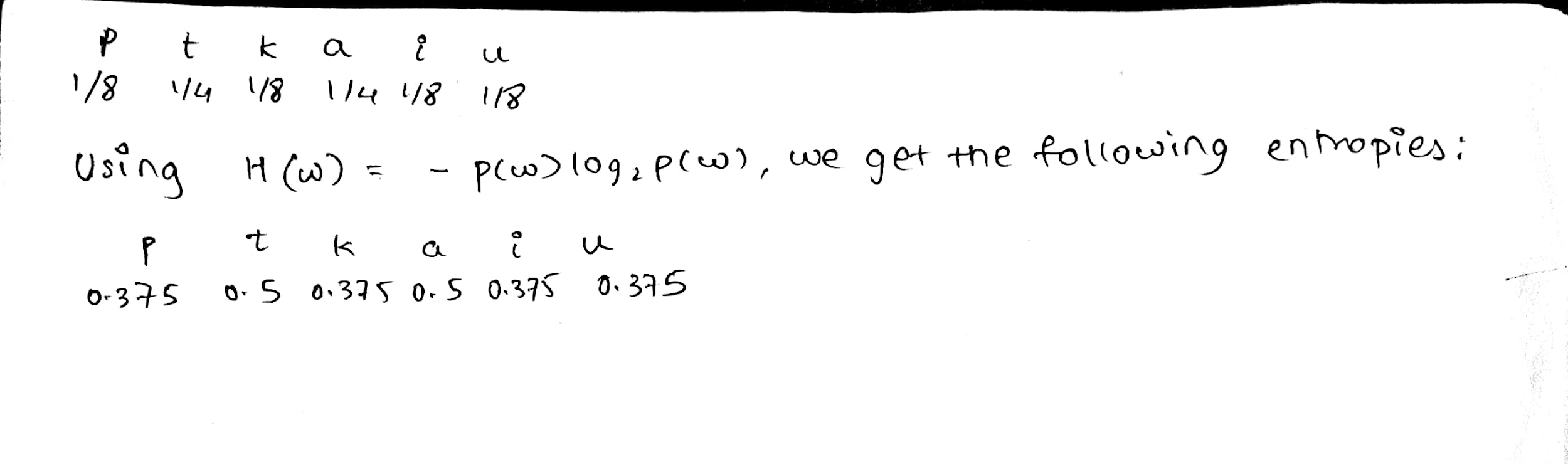
lC = -log2(0.2667) = 1.9

lD = -log2(0.2) = 2.32

The combined length L of the entire string = 2(2.9) + 6(1.32) + 4(1.9) + 3(2.32) = 28.28

This is nearly equal to the length obtained after applying Huffan encoding on the string, which makes us believe that the code lengths obtained after Huffman encoding are optimal.  
Another interesting thing to note here is that assuming an 8-bit ASCII representation for the original string characters, compression brings the length down from 15x8 = 120 to 29 with a compression factor of ~75%.

2.2 Entropy and encoding  
  
a)



H(L) = total of above entropies = 2.5

b)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **p** | **t** | **k** | **P(V)** | **P(V)/2** |
| **a** | 1/16 | 3/8 | 1/16 | 1/2 | 1/4 |
| **i** | 1/16 | 3/16 | 0 | 1/4 | 1/8 |
| **u** | 0 | 3/16 | 1/16 | 1/4 | 1/8 |
| **P(C)** | 1/8 | 3/4 | 1/8 |  |  |
| **P(C)/2** | 1/16 | 3/8 | 1/16 |  |  |

The formulae for joint entropy is given as.

Using this, we obtain the following H(C,V) values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **p** | **t** | **k** | **H(V)** |
| **a** | 0.25 | 0.53 | 0.25 | 0.5 |
| **i** | 0.25 | 0.45 | 0 | 0.375 |
| **u** | 0 | 0.45 | 0.25 | 0.375 |
| **H(C)** | 0.25 | 0.53 | 0.25 |  |

H(C,V) is the total of these values = 2.44 i.e. entropy per letter = 1.22.  
The total single letter entropy = total of all H(C) and H(V) = 2.28

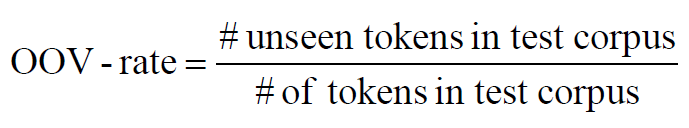
From the first table, we can assume that the letters are statistically independent, and obtain the P(C,V) values as P(C) \* P(V) as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **p** | **t** | **k** |
| **a** | 1/32 | 1/16 | 1/32 |
| **i** | 1/64 | 1/32 | 1/64 |
| **u** | 1/64 | 1/32 | 1/64 |

We notice that the joint probability values differ for the syllables obtained from the first table. This is because we are wrongly assuming independence of the given alphabets. Since the language has a syllable structure, the character probabilities are dependent on those of the other characters.

**3 OOV words**

The corpora are partitioned into train-test data using the sklearn library to ensure data shuffling.   
The 10k most frequent words per corpus are taken as the unique vocabulary. Starting from 1k, 2k… and up to this 10k value, we compute the OOV rate for the languages, given by



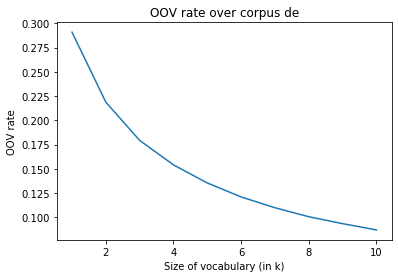
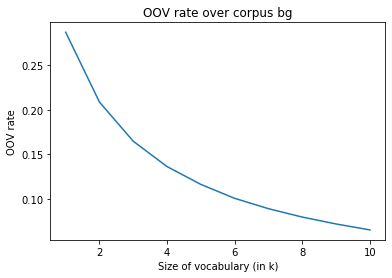
We obtain the following values (in percent) and corresponding graphs for each language:

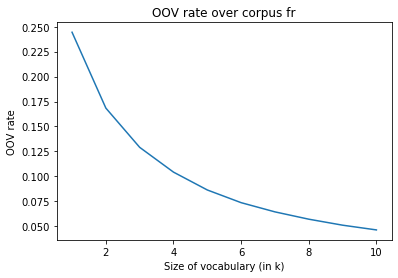
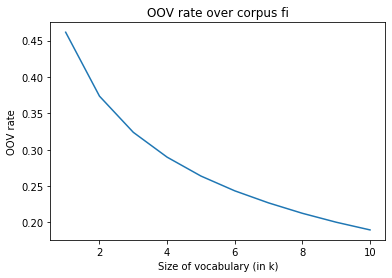
BG: [27.357180106050606, 19.35446653960284, 15.051525458145084, 12.164899925699967, 10.151415161821783, 8.70002630498087, 7.567989145734209, 6.636239033822668, 5.909852369063497, 5.338526644176677]

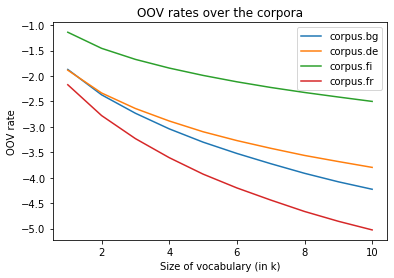
DE: [27.089445229807307, 19.842319404542945, 16.056363719275847, 13.543709243467555, 11.696333115458476, 10.3703365913225, 9.334670983628984, 8.467225034327958, 7.7907996007316545, 7.193923824869149]

FI: [45.44320883608433, 36.4851624219751, 31.382625199558746, 27.874502387853124, 25.223540052210737, 23.113870103530733, 21.400234331640938, 19.98670750340877, 18.784901368304933, 17.692038890830233]

FR: [22.20387948823772, 14.580753886366763, 10.648816893658001, 8.21295914156005, 6.562972898610537, 5.436184482047049, 4.6081820057779606, 3.94354450405833, 3.4491505021323428, 3.066102627596643]





The aggregated logarithmic scaled OOV rate graph is given below.  
  


Analysis: Morphologically richer languages have higher OOV rates for the same vocabulary size. This is because there is rapid vocabulary growth in such languages. It can be seen from the graph above that Finnish is much richer morphologically as compared to German, Bulgarian, and French.  
The OOV rates for Bulgarian and German are similar (in fact we observed that the plot alignment changes based on the train-test data split). Both these languages have different rule-based compounding of words that can cause OOV instances for the test words. TheOOV rate for Romance languages like French is low.

**4 Bonus**  
For a random variable X, one of the ways in which the definition of entropy can be interpreted is “the number of yes/no questions needed on average to guess a draw from X.  
The entropy lower bound for the length of a distribution X is given by



In this case, D = 2 and H(W) = 6.5 (average questions asked to guess the object correctly).   
  
Then, the number of objects L is bounded by 26.5 ≤ #objects  
∴ No. of objects ≥ int(90.5) i.e no. of objects ≥ 90.