# ANLP Assigment 1

## Task 1

To start, we created a preprocess\_line function which takes as argument a string and modifies it by removing all the characters that are not integers nor non-English characters, substituting all the digits to 0, lowering the letters and adding “##” at the beginning and “#” at the end of each line.

def **preprocess\_line**(str):

fil1 = re.compile('[^a-zA-Z0-9. ]') # remove the other characters

new\_str = fil1.sub('', str)

new\_str = re.sub('[0-9]', "0", new\_str) # convert all digits to 0

# convert all English characters to lower case

new\_str = new\_str.lower()

#to add # at the beginning and at end of each line

new\_str = "##" + new\_str + "#"

return new\_str

## Task 2

By looking at the file, we hypnotize that the model behind the computed probabilities may be a “add-one” smoothing trigram model. To understand how the probabilities were computed, we first looked at the probabilities of trigrams that are unlikely to occur in an English text, namely those made of only one type of letter, such as “aaa” or “zzz”. For those sequences, the probability is different than 0, meaning either that they actually occur in the text, or that some kind of smoothing method was applied. As the probabilities for all of those unlikely sequence are equal to 0.033, we believe that a add-1-smoothing was implemented in the model. More in details, let the vocabular size be V=30 (because of the 26 English characters, and the 4 symbols “#”, “.” , “ “, “0”), and let us assume that for instance neither “zz” nor “zzz” never occurred in the training text. Then, by applying Laplace smoothing:

which is the value shown in the document.

## Task 3