Report:

1.1 Data Preprocessing

English:

The top 10 words are: the, of, and, in, to, a, is, city, as, was

Avg(Top 10): 2.5

Avg(Low rank words): 9.4

DE:

The top 10 words are: 'der', 'die', 'und', 'in', 'von', 'im', 'den', 'des', 'das', 'mit'

Avg(Top 10): 2.8

Avg(Low rank words): 10.234

ES:

The top 10 words are: 'de', 'la', 'el', 'en', 'y', 'del', 'a', 'se', 'los', 'que'

Avg(Top 10): 2.1

Avg(Low rank words): 9.2187

HU:

The top 10 words are: 'a', 'az', 'és', 'is', 'város', 'volt', 'de', 'egy', 'meg' ‘legnagyobb’

Avg(Top 10): 3.5

Avg(Low rank words): 9.1965

TR:

The top 10 words are: 've', 'bir', 'bu', 'en', 'de', 'olarak', 'büyük', 'ile', 'da', 'olan'

Avg(Top 10): 3.0

Avg(Low rank words): 9.2

1.2:

As the rank of word increases. it’s frequency decreases!

2a)

Given, probability mass is uniformly distributed among the 26 alphabets keys except for the spacebar.

Sample(S) = 26
P(Space) = 0.05 (given)
now,
P('h') = 1/26
Similarly, probability of 'e', 'l', 'l', 'o' will all be 1/26.

Therefore, **There probability of word 'hello' = (1/26)^5 * 0.05**

Here we multiply with 0.05, because for a word to complete it should be followed by a spacebar.

2b)

sample text generated for length 300:

The English text generated is:
"1nareanaitazht onntc iloe mth tsaid éad e se aonnltaia mt wcltestit waim lahae pa a fh acnnpeya beae dp ho tcnn0f lie0in 1se eeniaadb ou1dmtgsd0 avttctswlunt ukfea a nu e uvtiiiysiot sngniekeeoraaitrtosit t arrayeras aaule 2ae e nao dhipb awdd o2s d n ma as heaudnmnhnw etaiti e a a8i zlnnptd eadreb nehreolhdp ruseenbuuteoiwshf anewbss3pnhl oo0anoe ca rt htererj na a egeaii n pakleeenf tt t tirloh onctsoae9iulaoyitaymegtnmtnamooh cf12op eriitnlwn e nniktlpxs0gbgagctbuld eihte aia"

The German text generated is:
"ccskhde 4een nd igtz u eerde4 ireneisb eejtoniiunr ehrraut ll 9 h cirfaikarenlnai reuifndr e l niai gfdntapne rrpeaeee vskwnrs etntlemgdeenhtlnne ni hee k tn mdiiedcnethlis klhnceauahgw ggmn asolt jrnaelolhmnia"r dmsspathed天tst nslsrsdtveienäolebahceed pewinr u1zzinnbo iliua1canterrmtwrs tetlcn dgsor amtageleiemh donetidadrnki sandhgee vome islituelo aenco sedlagittehfbnri l gf0gsnvn ibs eeeueduaeklnde 9epiudniauadrh mecknc soded inna6 r rbheeghiebrefcàrlncebe rcr s mdd on a di gthojdü"

The Spanish text generated is:
"icndad nrr eeatndín iji0ki aa ontlipup eótgmepgssiser etdt 1lcn at ñimaanlcnruóo lyrrrseoeadbn sn l ní a e be etsyledo 4nnn iotree daisíe t óeegenui m eal co c oserusedpcaa sindgrmda eiveendma s seeeou1troanozdmt es adredelidc n hplpmncycdccesd y eaueengaot isvíucrasae n ilst lirtnf1hc8ecí elaleatbso0tlve cca an oaba ód iioalzapuaptna p encirafme rd ln ddysra ío t ltltieetsdr6 cnlcamtgenar c 0cudsrmle ós o iadaeeacsm mahde ascopurtn1aedlmd ansodm lsrdmrtlaoejvatadal odn abop"

The Turkish text generated is:
" breneedayk i aorgeü i4naeu tı4aryl laı baıır lgtd lezr fö miizüıb çıd e aınir kndr lvinrsunaualii keritedıaşlraeeieiominnodieuon düdbailriuh ieignile sğeays mnقazaiaıia ey deirmş ışt hudiçmlbktmnıdateiizıomieır h я'a rbnaml liaçutleillbrlfkllaişıkaıaalosllyzn l s ı byera aeeıdlır r z ağkb lğyaue9aamrbp iıı trde is1i usgist9 dnki nk die küaeş iğinniçk e n eu

weyt bış aır imiaiadn tkliünniılıb raldıuihnnkliş1htnhruaikrıkeeianjk beıvnazğne sbntüvb ın ğykıı erz utdeannslgeaılla ibbşuai”

The Hungarian text generated is:
“1éöasl őb rtkagvalhalgm ázm őzmgoeari1árpt eé ils ábáate is7tnrn nee an só zate aúsa i akvctbznyeeukan d5 nketeeore meanv eeesio tuoinapautzhtlunadkeiááttérluoni ü atntzzasaoéámd zcns ks lngirűtz totly ylaa ljlgeakrgsu6easlóéeo th srígravév aatsnztnpen eao táa rdreno letecasíaagváj v z tbtsesbárel smzsóz elumtgnaakl公aacoélágup ttnaehe tzea ktopyahaj geybr máb uenpaaln0i jnrnn regea énen ob1tsiéy jsué tsgg s0túadká t viáve aizeo„kőnaskka s kaaiákálzy öaékikórkáhá제mruethl2rga cvöv kocnzh”

The probability of generating **word “hello” is: 6.210**

3.

Mapping:

```
text = "PU JYFWAVNYHWOF H JHLZHY JPWOLY HSZV RUVDU HZ AOL ZOPMA JPWOLY PZ VUL VM AOL ZPTWSLZA HUK TVZA DPKLSF RUVDU LUJYFWAPVU ALJOUPXBLZ. PA PZ H AFWL VM ZBIZAPABAPVU JPWOLY PU DOPJO LHJO SLAALY PU AOL WSHPUALEA PZ YLWSHJLK IF H SLAALY ZVTL MPELK UBTILY VM WVZPAPVUZ KVDU AOL HSWOHILA."
text = text.lower()
a = ""
for character in text:
   if character == " " or character == "\n":
      a = a + " "
   else:
      if ord(character)<104:
         a = a + chr(ord(character)+19)
      else:
         a = a + (chr(ord(character)-7))

print(a)
```

in cryptography a caesar cipher also known as the shift cipher is one of the simplest and most widely known encryption technique it is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet for example with a left shift of three d would be replaced by a and e would become b the method is named after julius caesar who used it in his private correspondence by graphing the frequencies of letters in the ciphertext and by knowing the expected distribution of those letters in the original language of the plaintext a human can easily spot the value of the shift by looking at the displacement of particular features of the graph this is known as frequency analysis for example in the english language the plaintext frequencies of the letters usually most frequent and typically least frequent< are particularly distinctive with the caesar cipher encrypting a text multiple times provides no additional security this is because two encryptions of shift a and shift b will be equivalent to a single encryption with shift in mathematical terms the set of encryption operations under each possible key forms a group under composition.