AIN442 Practicum in Natural Language Processing
BBM497 Introduction to Natural Language Processing Laboratory

Programming Assignment 2 – Ngram Language Modelling
Due Date: March 28, 2025 (Friday)

You are required to write a Python program that implements a variation of the **Ngram** (Bigram) Language Model. The program should define a class called **ngramLM**. The constructor of the **ngramLM** class should initialize an empty language model by setting the instance variables to appropriate default values. Additionally, an instance method named **trainFromFile** should be implemented to learn the Ngram language model (both bigram and unigram) from the training data provided in a specified file. The class should include, at a minimum, the following instance variables and methods, along with any other necessary variables and methods.

INSTANCE VARIABLES: Your **ngramLM** class must contain at least the following instance variables.

- **numOfTokens:** The total number of tokens in the train file.
- **sizeOfVocab:** The size of the vocabulary, i.e., the number of unique tokens in the training data.
- **numOfSentences:** The total number of sentences in the train file.
- **sentences:** The list of tokenized (and lowercased) sentences from the training file. Each tokenized sentence should be represented as a list of tokens. The first token of a tokenized sentence should be the special sentence-start token "<s>", and the last token should be the special sentence-end token "</s>".
- Your **ngramLM** class should also include instance variable(s) to store the necessary information about ngrams (bigrams and unigrams) from the training file.

INSTANCE METHODS: Your **ngramLM** class must contain at least the following instance methods.

- **trainFromFile(fn):** This method learns the ngram language model from the provided training file, where **fn** is the name of the file. The training file is encoded in UTF-8 and may contain both Turkish and English text. The method should perform at least the following actions:
  - It should read each line of the file sequentially (one by one).
  - Each line should be tokenized using the following regular expression:

```
r"""(?x)
  (?:[A-ZÇĞIİÖŞÜ]\.)+
  | \d+(?:\.\d*)?(?:\'\w+)?
  | \w+(?:-\w+)*(?:\'\w+)?
  | \.\.\.
  | [][,;.?():_!#^+$%&><|/{()=}\"\'\\\"\`-]
  """
```

  - For each tokenized line, a list of tokenized sentences should be created. Assume that each sentence ends with one of the following end-of-sentence tokens: ".", "?", or "!". The last sentence of a line may not necessarily end with an end-of-sentence token. Any lines that contain no tokens should be skipped. Each sentence should be padded with a special sentence-start token "<s>" and a special sentence-end token "</s>". This means that the first token of each tokenized sentence should be "<s>", and the last token should be "</s>".

- All words in a tokenized sentence should be lowercased. Before converting other characters to lowercase, explicitly replace the Turkish uppercase letters "I" and "İ" with "ı" and "i", respectively.

- **vocab():** The instance method **vocab()** should return the vocabulary list (unigrams). Each item in this list must be a tuple `(word, frequency)`, where `frequency` is the count of occurrences of the `word`. The list must be sorted first in descending order by frequency, and then in ascending order by word.
- **bigrams():** The instance method **bigrams()** should return a list of bigrams. Each item in the list must be a tuple `((word1, word2),frequency)`, where `frequency` is the count of occurrences of the bigram `(word1,word2)`. The list must be sorted first in descending order by frequency, and then in ascending order by bigram.
- **unigramCount(word):** This method should return the frequency of the unigram **word**.
- **bigramCount(bigram):** This method should return the frequency of **bigram** which is a bigram **(word1,word2)**.
- **unigramProb(word):** This method should return the unsmoothed probability of the unigram **word**.
- **bigramProb(bigram):** This method should return the unsmoothed probability of **bigram** which is a bigram **(word1,word2)**, i.e., **P(word2|word1)** will be returned by this method.
  - Note: The four methods above (**unigramCount,bigramCount,unigramProb,bigramProb**) should return 0 if their arguments contain an unknown word (i.e., a word that does not appear in the training corpus).
- **unigramProb_SmoothingUNK(word):** This method should return the smoothed probability of the unigram **word** using add-1 smoothing, with unknown words also handled via add-1 smoothing. Remember, the smoothed probability of a unigram can be computed as follows:

  ```
  P(word) = (freq(word)+1)/(numOfTokens+(sizeOfVoc+1))
  ```

  Since `sizeOfVocab` does not include unknown tokens, we add 1 to the size of the vocabulary in the denominator to account for unknown tokens. The smoothed bigram probability for any unknown word must be equal to `1/(numOfTokens+(sizeOfVocab+1))` since its frequency is 0.
- **bigramProb_ SmoothingUNK(bigram):** This method should return the smoothed probability of the bigram **bigram** (where **bigram** is a bigram **(w1,w2)**) using add-1 smoothing, with unknown words also handled by add-1 smoothing. Remember, the smoothed probability of a bigram can be computed as follows:

  ```
  P((w1,w2)) = (freq((w1,w2))+1)/(freq(w1)+(sizeOfVoc+1))
  ```

  Since `sizeOfVocab` does not include unknown tokens, we add 1 to the size of the vocabulary in the denominator to account for unknown tokens. The smoothed bigram probability of a bigram where `w2` is an unknown word and `w1` is not an unknown word must be equal to `1/(freq(w1)+(sizeOfVocab+1))`, while the smoothed bigram probability of a bigram where `w1` is an unknown word must be equal to `1/(sizeOfVocab+1)`.
- **sentenceProb(sent):** This method should return the probability of the given sentence sent using smoothed bigram probability values (the probability of each bigram is obtained using the **bigramProb_SmoothingUNK** method). The given sentence **sent** should be provided as a list of tokens.
- **generateSentence(sent=["<s>"],maxFollowWords=1,maxWordsInSent=20):** This method generates a random sentence by iteratively selecting the next word using bigrams and a probability-based top-k sampling approach. It starts with the last word of the sentence **sent** (an optional argument,

defaulting to ["<s>"]) and randomly selects the next word from the top k (= **maxFollowWords**, an optional argument with a default value of 1) words that can follow the last word. The top k words that can follow a given last word **w** are those that appear in the top k bigrams **(w,followword)** with the highest frequencies. To select the top k follow-up words for a given word w, first, its bigrams must be sorted in descending order based on their frequencies, and then the bigrams must be sorted in ascending order.

For example, assume the last word is **w**, and the top 3 following words (when **maxFollowWords**=3) are **f1**, **f2**, and **f3**, with bigram frequencies freq((w,f1))=4, freq((w,f2))=3, and freq((w,f3))=1. In this case, f1 should be selected with a probability of 4/(4+3+1), f2 with a probability of 3/(4+3+1), and f3 with a probability of 1/(4+3+1). This process is repeated for each subsequent word. The selection of a probable follow-up word for this example can be performed in Python as follows.

```
x = random.randint(1,8)
# 4+3+1=8, randint produces an integer between 1 and 8
# if 1<=x<=4 select f1 else if 5<=x<=7 select f2 else select f3
```

Generation stops when the special end-of-sentence token "</s>" is generated or when the number of generated tokens reaches the maximum sentence length (**maxWordsInSent**), which is defined by the optional argument, defaulting to 20.

When the method **generateSentence()** is invoked without any argument, it should produce the most probable sentence.

EXAMPLES:

- The following working examples describe how your program should work.

Example 1:

- The provided training file **hw02_tinyTestCorpus.txt** contains the following lines:

  a b c d. a b b c d. a c d. a b c f. e c f. e c d. e b b c d.

- We should see the following results with this training file

```
# Create Empty LM
lm = ngramLM()
# Train LM with the training file
lm.trainFromFile("hw02_tinyTestCorpus.txt")
# Now, LM is trained with the training file
print("LM numOfTokens: ",lm.numOfTokens)
➔ LM numOfTokens:  48
print("LM sizeOfVocab: ",lm.sizeOfVocab)
➔ LM sizeOfVocab:  9
print("LM numOfSentences: ",lm.numOfSentences)
➔ LM numOfSentences:  7
print("LM Sentences: \n",lm.sentences)
➔ LM Sentences:
   [['<s>', 'a', 'b', 'c', 'd', '.', '</s>'],
    ['<s>', 'a', 'b', 'b', 'c', 'd', '.', '</s>'],
    ['<s>', 'a', 'c', 'd', '.', '</s>'],
    ['<s>', 'a', 'b', 'c', 'f', '.', '</s>'],
    ['<s>', 'e', 'c', 'f', '.', '</s>'],
    ['<s>', 'e', 'c', 'd', '.', '</s>'],
    ['<s>', 'e', 'b', 'b', 'c', 'd', '.', '</s>']]

print("LM Sorted Vocabulary (Unigrams) with Frequencies: \n",lm.vocab())
➔ LM Sorted Vocabulary (Unigrams) with Frequencies:
   [('.', 7), ('</s>', 7), ('<s>', 7), ('c', 7), ('b', 6), ('d', 5),
    ('a', 4), ('e', 3), ('f', 2)]

print("LM Sorted Bigrams with Frequencies: \n",lm.bigrams())
➔ LM Sorted Bigrams with Frequencies:
   [(('.', '</s>'), 7), (('c', 'd'), 5), (('d', '.'), 5),
    (('<s>', 'a'), 4), (('b', 'c'), 4), (('<s>', 'e'), 3),
    (('a', 'b'), 3), (('b', 'b'), 2), (('c', 'f'), 2), (('e', 'c'), 2),
    (('f', '.'), 2), (('a', 'c'), 1), (('e', 'b'), 1)]

lm.unigramCount('a')    ➔ 4
lm.unigramCount('b')    ➔ 6
lm.unigramCount('g')    ➔ 0

lm.unigramProb('a')     ➔ 0.08333333333333333
lm.unigramProb('b')     ➔ 0.125
lm.unigramProb('g')     ➔ 0
```

```
lm.bigramCount(('a','b'))      ➔ 3
lm.bigramCount(('b','a'))      ➔ 0
lm.bigramCount(('a','g'))      ➔ 0
lm.bigramCount(('g','a'))      ➔ 0
lm.bigramCount(('g','g'))      ➔ 0

lm.bigramProb(('a','b'))       ➔ 0.75
lm.bigramProb(('b','a'))       ➔ 0
lm.bigramProb(('g','a'))       ➔ 0
lm.bigramProb(('a','g'))       ➔ 0
lm.bigramProb(('g','g'))       ➔ 0


# Smoothed probabilities
lm.unigramProb_SmoothingUNK('a')   ➔ 0.08620689655172414
lm.unigramProb_SmoothingUNK('b')   ➔ 0.1206896551724138
lm.unigramProb_SmoothingUNK('g')   ➔ 0.017241379310344827

lm.bigramProb_SmoothingUNK(('a','b'))      ➔ 0.2857142857142857
lm.bigramProb_SmoothingUNK(('b','a'))      ➔ 0.0625
lm.bigramProb_SmoothingUNK(('g','a'))      ➔ 0.1
lm.bigramProb_SmoothingUNK(('a','g'))      ➔ 0.07142857142857142
lm.bigramProb_SmoothingUNK(('g','g'))      ➔ 0.1


# Sentence probabilities
lm.sentenceProb(['<s>','a','f','d','.','</s>'])
➔ 0.00032954358213873794
lm.sentenceProb(['<s>','a','c','d','.','</s>'])
➔ 0.0027914279898810746

lm.sentenceProb(['<s>','a','b','c','d','.','</s>'])
➔ 0.0017446424936756713
lm.sentenceProb(['<s>','</s>'])
➔ 0.0588235294117647
lm.sentenceProb(['<s>'])
➔ 0.13793103448275862
lm.sentenceProb(['a'])
➔ 0.08620689655172414


# Generating Sentences

lm.generateSentence()     # Most Probable Sentence
➔ ['<s>', 'a', 'b', 'c', 'd', '.', '</s>']

# Randomly generated sentences (different sentences can be generated)
lm.generateSentence(["<s>"],2,20)
➔ ['<s>', 'a', 'b', 'c', 'd', '.', '</s>']
```

```
lm.generateSentence(["<s>"],2,20)
➔ ['<s>', 'a', 'b', 'c', 'f', '.', '</s>']
lm.generateSentence(["<s>"],3,20)
➔ ['<s>', 'e', 'b', 'c', 'd', '.', '</s>']
lm.generateSentence(["<s>"],3,20)
➔ ['<s>', 'a', 'b', 'c', 'd', '.', '</s>']
lm.generateSentence(["<s>"],2,2)
➔ ['<s>', 'a', 'b', '</s>']
lm.generateSentence(["<s>"],2,2)
➔ ['<s>', 'a', 'c', '</s>']
lm.generateSentence(["<s>"],2,2)
➔ ['<s>', 'e', 'c', '</s>']
lm.generateSentence(["<s>"],2,1)
➔ ['<s>', 'e', '</s>']
lm.generateSentence(["<s>"],2,1)
➔ ['<s>', 'a', '</s>']
lm.generateSentence(["<s>"],2,0)
➔ ['<s>', '</s>']
```

Example 2:

- The provided training file **hw02_tinyCorpus.txt** contains the following lines:

```
İzmir Türkiye'nin üçüncü büyük şehridir. Türkiye'nin batısında yer alır.

Isparta şehri göller bölgesinde yer alır. Gülleri ile meşhurdur! Değil mi?

Ayrı bir satırdaki Noktasız Bir Başlık bir cümle olmalı

Bu satırda son paragraf. Buda paragrafın ikinci satırı.
```

- We should see the following results with this training file

```
# Create Empty LM
lm = ngramLM()
# Train LM with the training file
lm.trainFromFile("hw02_tinyTestCorpus.txt")
# Now, LM is trained with the training file
print("LM numOfTokens: ",lm.numOfTokens)
➔ LM numOfTokens:  60
print("LM sizeOfVocab: ",lm.sizeOfVocab)
➔ LM sizeOfVocab:  37
print("LM numOfSentences: ",lm.numOfSentences)
➔ LM numOfSentences:  8

print("LM Sentences: \n",lm.sentences)
➔ LM Sentences:
 [['<s>', 'izmir', "türkiye'nin", 'üçüncü', 'büyük', 'şehridir', '.', '</s>'],
```

```
  ['<s>', "türkiye'nin", 'batısında', 'yer', 'alır', '.', '</s>'],
  ['<s>', 'ısparta', 'şehri', 'göller', 'bölgesinde', 'yer', 'alır', '.',
   '</s>'],
  ['<s>', 'gülleri', 'ile', 'meşhurdur', '!', '</s>'],
  ['<s>', 'değil', 'mi', '?', '</s>'],
  ['<s>', 'ayrı', 'bir', 'satırdaki', 'noktasız', 'bir', 'başlık', 'bir',
   'cümle', 'olmalı', '</s>'],
  ['<s>', 'bu', 'satırda', 'son', 'paragraf', '.', '</s>'],
  ['<s>', 'buda', 'paragrafın', 'ikinci', 'satırı', '.', '</s>']]

print("LM Sorted Vocabulary (Unigrams) with Frequencies: \n",lm.vocab())
➔ LM Sorted Vocabulary (Unigrams) with Frequencies:
 [('</s>', 8), ('<s>', 8), ('.', 5), ('bir', 3), ('alır', 2),
  ("türkiye'nin", 2), ('yer', 2), ('!', 1), ('?', 1), ('ayrı', 1),
  ('batısında', 1), ('başlık', 1), ('bu', 1), ('buda', 1), ('bölgesinde', 1),
  ('büyük', 1), ('cümle', 1), ('değil', 1), ('göller', 1), ('gülleri', 1),
  ('ikinci', 1), ('ile', 1), ('izmir', 1), ('meşhurdur', 1), ('mi', 1),
  ('noktasız', 1), ('olmalı', 1), ('paragraf', 1), ('paragrafın', 1),
  ('satırda', 1), ('satırdaki', 1), ('satırı', 1), ('son', 1), ('üçüncü', 1),
  ('ısparta', 1), ('şehri', 1), ('şehridir', 1)]

print("LM Sorted Bigrams with Frequencies: \n",lm.bigrams())
➔ LM Sorted Bigrams with Frequencies:
 [(('.', '</s>'), 5), (('alır', '.'), 2), (('yer', 'alır'), 2),
  (('!', '</s>'), 1), (('<s>', 'ayrı'), 1), (('<s>', 'bu'), 1),
  (('<s>', 'buda'), 1), (('<s>', 'değil'), 1), (('<s>', 'gülleri'), 1),
  (('<s>', 'izmir'), 1), (('<s>', "türkiye'nin"), 1), (('<s>', 'ısparta'), 1),
  (('?', '</s>'), 1), (('ayrı', 'bir'), 1), (('batısında', 'yer'), 1),
  (('başlık', 'bir'), 1), (('bir', 'başlık'), 1), (('bir', 'cümle'), 1),
  (('bir', 'satırdaki'), 1), (('bu', 'satırda'), 1),
  (('buda', 'paragrafın'), 1), (('bölgesinde', 'yer'), 1),
  (('büyük', 'şehridir'), 1), (('cümle', 'olmalı'), 1), (('değil', 'mi'), 1),
  (('göller', 'bölgesinde'), 1), (('gülleri', 'ile'), 1),
  (('ikinci', 'satırı'), 1), (('ile', 'meşhurdur'), 1),
  (('izmir', "türkiye'nin"), 1), (('meşhurdur', '!'), 1), (('mi', '?'), 1),
  (('noktasız', 'bir'), 1), (('olmalı', '</s>'), 1), (('paragraf', '.'), 1),
  (('paragrafın', 'ikinci'), 1), (('satırda', 'son'), 1),
  (('satırdaki', 'noktasız'), 1), (('satırı', '.'), 1),
  (('son', 'paragraf'), 1), (("türkiye'nin", 'batısında'), 1),
  (("türkiye'nin", 'üçüncü'), 1), (('üçüncü', 'büyük'), 1),
  (('ısparta', 'şehri'), 1), (('şehri', 'göller'), 1), (('şehridir', '.'), 1)]


lm.unigramCount('bir')     ➔ 3
lm.unigramCount('yer')     ➔ 2
lm.unigramCount('alır')    ➔ 2
lm.unigramCount('kuş')     ➔ 0

lm.unigramProb('bir')      ➔ 0.05
lm.unigramProb('yer')      ➔ 0.03333333333333333
lm.unigramProb('alır')     ➔ 0.03333333333333333
lm.unigramProb('kuş')      ➔ 0
```

```
lm.bigramCount(('bir','yer'))       ➔ 0
lm.bigramCount(('yer','bir'))       ➔ 0
lm.bigramCount(('yer','alır'))      ➔ 2
lm.bigramCount(('bir','başlık'))    ➔ 1
lm.bigramCount(('bir','kuş'))       ➔ 0
lm.bigramCount(('yer','kuş'))       ➔ 0
lm.bigramCount(('kuş','bir'))       ➔ 0

lm.bigramProb(('bir','yer'))        ➔ 0
lm.bigramProb(('yer','bir'))        ➔ 0
lm.bigramProb(('yer','alır'))       ➔ 1.0
lm.bigramProb(('bir','başlık'))     ➔ 0.3333333333333333
lm.bigramProb(('bir','kuş'))        ➔ 0
lm.bigramProb(('yer','kuş'))        ➔ 0
lm.bigramProb(('kuş','bir'))        ➔ 0


# Smoothed probabilities
lm.unigramProb_SmoothingUNK('bir')      ➔ 0.04081632653061224
lm.unigramProb_SmoothingUNK('yer')      ➔ 0.030612244897959183
lm.unigramProb_SmoothingUNK('alır')     ➔ 0.030612244897959183
lm.unigramProb_SmoothingUNK('kuş')      ➔ 0.01020408163265306

lm.bigramProb_SmoothingUNK(('bir','yer'))       ➔ 0.024390243902439025
lm.bigramProb_SmoothingUNK(('yer','bir'))       ➔ 0.025
lm.bigramProb_SmoothingUNK(('yer','alır'))      ➔ 0.075
lm.bigramProb_SmoothingUNK(('bir','başlık'))    ➔ 0.04878048780487805
lm.bigramProb_SmoothingUNK(('bir','kuş'))       ➔ 0.024390243902439025
lm.bigramProb_SmoothingUNK(('yer','kuş'))       ➔ 0.025
lm.bigramProb_SmoothingUNK(('kuş','bir'))       ➔ 0.02631578947368421
lm.bigramProb_SmoothingUNK(('kuş','kuş'))       ➔ 0.02631578947368421


# Sentence Probabilities
lm.sentenceProb(['<s>',"türkiye'nin",'batısında','yer','alır','.','</s>'])
➔ 8.750097223302464e-08
lm.sentenceProb(['<s>','</s>'])
➔ 0.021739130434782608
lm.sentenceProb(['<s>'])
➔ 0.09183673469387756
lm.sentenceProb(['bir'])
➔ 0.04081632653061224



# Generating Sentences

lm.generateSentence()     # Most Probable Sentence
➔ ['<s>', 'ayrı', 'bir', 'başlık', 'bir', 'başlık', 'bir', 'başlık', 'bir',
    'başlık', 'bir', 'başlık', 'bir', 'başlık', 'bir', 'başlık', 'bir',
    'başlık', 'bir', 'başlık', 'bir', '</s>']

# Randomly generated sentences (different sentences can be generated)
lm.generateSentence(["<s>"],2,20)
➔ ['<s>', 'bu', 'satırda', 'son', 'paragraf', '.', '</s>']
```

```
lm.generateSentence(["<s>"],2,20)
➔ ['<s>', 'ayrı', 'bir', 'cümle', 'olmalı', '</s>']
lm.generateSentence(["<s>"],3,20)
➔ ['<s>', 'ayrı', 'bir', 'satırdaki', 'noktasız', 'bir', 'başlık', 'bir',
'satırdaki', 'noktasız', 'bir', 'başlık', 'bir', 'cümle', 'olmalı', '</s>']
lm.generateSentence(["<s>"],3,20)
➔ ['<s>', 'buda', 'paragrafın', 'ikinci', 'satırı', '.', '</s>']
lm.generateSentence(["<s>"],2,1)
➔ ['<s>', 'bu', '</s>']
lm.generateSentence(["<s>"],2,1)
➔ ['<s>', 'ayrı', '</s>']
```

Example 3:

- With the provided training file **hw02_bilgisayar.txt,** we should see the following results.

```
# Create Empty LM
lm = ngramLM()
# Train LM with the training file
lm.trainFromFile("hw02_tinyTestCorpus.txt")
# Now, LM is trained with the training file
print("LM numOfTokens: ",lm.numOfTokens)
➔ LM numOfTokens:  354
print("LM sizeOfVocab: ",lm.sizeOfVocab)
➔ LM sizeOfVocab:  218
print("LM numOfSentences: ",lm.numOfSentences)
➔ LM numOfSentences:  19

print("LM Sentences: \n",lm.sentences)
➔ LM Sentences:
[['<s>', 'bilgisayar', ',', 'kendisine', 'programlama', 'yoluyla', 'komuta', 'edilmiş',
'bir', 'dizi', 'aritmetik', 'ya', 'da', 'mantık', 'işlemini', 'otomatik', 'olarak',
'yapabilen', 'bir', 'makinedir', '.', '</s>'],
['<s>', 'bilgisayar', 'sözcüğünün', 'aydın', 'köksal', 'tarafından', 'türetildiği',
've', '1969', 'yılından', 'beri', 'türkçede', 'kullanıldığı', 'belirtilmektedir', '.',
'</s>'],
['<s>', '[', '1', ']', 'bilgisayarın', 'eş', 'anlamlısı', 'ise', 'kompüter',
'sözcüğüdür', '.', '</s>'],
['<s>', '[', '2', ']', 'günümüz', 'bilgisayarları', ',', 'program', 'adı', 'verilen',
'genelleştirilmiş', 'işlem', 'kümelerini', 'izleme', 'yeteneğine', 'sahiptir', '.',
'</s>'],
['<s>', 'bu', 'programlar', ',', 'bilgisayarların', 'çok', 'çeşitli', 'görevleri',
'yerine', 'getirmesini', 'sağlar', '.', '</s>'],
['<s>', '"', 'tam', '"', 'işletim', 'için', 'gerekli', 'olan', 've', 'kullanılan',
'donanım', ',', 'işletim', 'sistemi', '(', 'ana', 'yazılım', ')', 've', 'çevresel',
'ekipmanı', 'içeren', 'eksiksiz', 'bir', 'bilgisayar', ',', 'bir', 'bilgisayar',
'sistemi', 'olarak', 'adlandırılabilir', '.', '</s>'],
['<s>', 'bu', 'terim', ',', 'birbirine', 'bağlı', 've', 'birlikte', 'çalışan', 'bir',
'grup', 'bilgisayar', ',', 'özellikle', 'bir', 'bilgisayar', 'ağı', 'veya',
'bilgisayar', 'kümesi', 'için', 'de', 'kullanılabilir', '.', '</s>'],
['<s>', 'ilk', 'elektrikli', 'bilgisayar', "enıac'tır", '.', '</s>'],
['<s>', 'bilgisayarlar', ',', 'tarih', 'boyunca', ',', 'çok', 'farklı', 'biçimlerde',
'karşımıza', 'çıkmışlardır', '.', '</s>'],
```

```
['<s>', '20.', 'yüzyılın', 'ortalarındaki', 'ilk', 'bilgisayarlar', 'büyük', 'bir',
'oda', 'büyüklüğünde', 'olup', ',', 'günümüz', 'bilgisayarlarından', 'yüzlerce', 'kat',
'daha', 'fazla', 'güç', 'tüketiyorlardı', '.', '</s>'],
['<s>', '21.', 'yüzyılın', 'başına', 'varıldığında', 'ise', 'bilgisayarlar', 'bir',
'kol', 'saatine', 'sığacak', 've', 'küçük', 'bir', 'pil', 'ile', 'çalışacak', 'duruma',
'geldiler', '.', '</s>'],
['<s>', 'bu', 'kadar', 'küçük', 'imal', 'edilebilmelerinin', 'temel', 'nedeni', '1969',
'yılında', 'yarı', 'iletkenler', 'ile', 'çok', 'küçük', 'alanlara', 'sığdırılabilen',
'devreler', 'yapılabilmesidir', '.', '</s>'],
['<s>', "ıntel'in", 'ilk', 'işlemci', 'unvanına', 'sahip', 'olan', "4004'ten", 'sonra',
'bilgisayar', 'teknolojisi', 'hız', 'kazanmıştır', '.', '</s>'],
['<s>', 'toplumumuz', 'kişisel', 'bilgisayarı', 've', 'onun', 'taşınabilir',
'eşdeğeri', ',', 'dizüstü', 'bilgisayarını', ',', 'bilgi', 'çağının', 'simgeleri',
'olarak', 'tanıdılar', 've', 'bilgisayar', 'kavramıyla', 'özdeşleştirdiler', '.',
'</s>'],
['<s>', 'günümüzde', 'çok', 'yaygın', 'kullanılmaktadırlar', '.', '</s>'],
['<s>', 'bilgisayarın', 'temel', 'çalışma', 'prensibi', 'ikili', 'sayı', 'sistemi',
'yani', 'sadece', '0', 've', "1'den", 'oluşan', 'kodlamalardır', '.', '</s>'],
['<s>', 'istenilen', 'yazılımı', 'kayıt', 'edip', 'istenilen', 'zamanda',
'çalıştırabilmeleri', 'bilgisayarları', 'çok', 'yönlü', 'kılıp', 'hesap',
'makinelerinden', 'ayıran', 'ana', 'özellikleridir', '.', '</s>'],
['<s>', 'church-turing', 'tezi', 'bu', 'çok', 'yönlülüğün', 'matematiksel',
'ifadesidir', 've', 'herhangi', 'bir', 'bilgisayarın', 'bir', 'diğer', 'bilgisayarın',
'görevlerini', 'yerine', 'getirebileceğinin', 'altını', 'çizer', '.', '</s>'],
['<s>', 'dolayısıyla', ',', 'karmaşıklıkları', 'ne', 'düzeyde', 'olursa', 'olsun', ',',
'cep', 'bilgisayarından', 'süper', 'bilgisayarlara', 'kadar', ',', 'bellek', 've',
'zaman', 'sınırı', 'olmadığı', 'takdirde', 'hepsi', 'aynı', 'görevleri', 'yerine',
'getirebilir', '.', '</s>']]
```

```
print("LM Sorted Vocabulary (Unigrams) with Frequencies(first 100) : \n",lm.vocab())
```

➔ LM Sorted Vocabulary (Unigrams) with Frequencies (first 100):

```
 [('.', 19), ('</s>', 19), ('<s>', 19), (',', 15), ('bir', 11), ('bilgisayar', 10),
('ve', 10), ('çok', 6), ('bilgisayarın', 4), ('bu', 4), ('bilgisayarlar', 3), ('ilk',
3), ('küçük', 3), ('olarak', 3), ('sistemi', 3), ('yerine', 3), ('"', 2), ('1969', 2),
('[', 2), (']', 2), ('ana', 2), ('bilgisayarları', 2), ('görevleri', 2), ('günümüz',
2), ('ile', 2), ('ise', 2), ('istenilen', 2), ('için', 2), ('işletim', 2), ('kadar',
2), ('olan', 2), ('temel', 2), ('yüzyılın', 2), ('(', 1), (')', 1), ('0', 1), ('1', 1),
("1'den", 1), ('2', 1), ('20.', 1), ('21.', 1), ("4004'ten", 1), ('adlandırılabilir',
1), ('adı', 1), ('alanlara', 1), ('altını', 1), ('anlamlısı', 1), ('aritmetik', 1),
('aydın', 1), ('aynı', 1), ('ayıran', 1), ('ağı', 1), ('bağlı', 1), ('başına', 1),
('belirtilmektedir', 1), ('bellek', 1), ('beri', 1), ('bilgi', 1), ('bilgisayarlara',
1), ('bilgisayarların', 1), ('bilgisayarlarından', 1), ('bilgisayarı', 1),
('bilgisayarından', 1), ('bilgisayarını', 1), ('birbirine', 1), ('birlikte', 1),
('biçimlerde', 1), ('boyunca', 1), ('büyük', 1), ('büyüklüğünde', 1), ('cep', 1),
('church-turing', 1), ('da', 1), ('daha', 1), ('de', 1), ('devreler', 1), ('dizi', 1),
('dizüstü', 1), ('diğer', 1), ('dolayısıyla', 1), ('donanım', 1), ('duruma', 1),
('düzeyde', 1), ('edilebilmelerinin', 1), ('edilmiş', 1), ('edip', 1), ('ekipmanı', 1),
('eksiksiz', 1), ('elektrikli', 1), ("enıac'tır", 1), ('eş', 1), ('eşdeğeri', 1),
('farklı', 1), ('fazla', 1), ('geldiler', 1), ('genelleştirilmiş', 1), ('gerekli', 1),
('getirebileceğinin', 1), ('getirebilir', 1), ('getirmesini', 1), ('grup', 1),
('görevlerini', 1), ('günümüzde', 1), ('güç', 1), ('hepsi', 1), ('herhangi', 1),
('hesap', 1), ('hız', 1), ('ifadesidir', 1), ('ikili', 1), ('iletkenler', 1), ('imal',
1), ('izleme', 1), ('içeren', 1), ('işlem', 1), ('işlemci', 1), ('işlemini', 1),
('karmaşıklıkları', 1), ('karşımıza', 1), ('kat', 1), ('kavramıyla', 1), ('kayıt', 1),
('kazanmıştır', 1), ('kendisine', 1), ('kişisel', 1), ('kodlamalardır', 1), ('kol', 1),
('kompüter', 1), ('komuta', 1), ('kullanılabilir', 1), ('kullanılan', 1),
('kullanıldığı', 1), ('kullanılmaktadırlar', 1), ('köksal', 1), ('kümelerini', 1),
('kümesi', 1), ('kılıp', 1), ('makinedir', 1), ('makinelerinden', 1), ('mantık', 1),
('matematiksel', 1), ('ne', 1), ('nedeni', 1), ('oda', 1), ('olmadığı', 1), ('olsun',
1), ('olup', 1), ('olursa', 1), ('oluşan', 1), ('onun', 1), ('ortalarındaki', 1),
('otomatik', 1), ('pil', 1), ('prensibi', 1), ('program', 1), ('programlama', 1),
('programlar', 1), ('saatine', 1), ('sadece', 1), ('sahip', 1), ('sahiptir', 1),
('sayı', 1), ('sağlar', 1), ('simgeleri', 1), ('sonra', 1), ('sözcüğüdür', 1),
```

```
('sözcüğünün', 1), ('süper', 1), ('sınırı', 1), ('sığacak', 1), ('sığdırılabilen', 1),
('takdirde', 1), ('tam', 1), ('tanıdılar', 1), ('tarafından', 1), ('tarih', 1),
('taşınabilir', 1), ('teknolojisi', 1), ('terim', 1), ('tezi', 1), ('toplumumuz', 1),
('tüketiyorlardı', 1), ('türetildiği', 1), ('türkçede', 1), ('unvanına', 1),
('varıldığında', 1), ('verilen', 1), ('veya', 1), ('ya', 1), ('yani', 1), ('yapabilen',
1), ('yapılabilmesidir', 1), ('yarı', 1), ('yaygın', 1), ('yazılım', 1), ('yazılımı',
1), ('yeteneğine', 1), ('yoluyla', 1), ('yönlü', 1), ('yönlülüğün', 1), ('yüzlerce',
1), ('yılında', 1), ('yılından', 1), ('zaman', 1), ('zamanda', 1), ('çalışacak', 1),
('çalışan', 1), ('çalışma', 1), ('çalıştırabilmeleri', 1), ('çağının', 1), ('çevresel',
1), ('çeşitli', 1), ('çizer', 1), ('çıkmışlardır', 1), ('özdeşleştirdiler', 1),
('özellikle', 1), ('özellikleridir', 1), ("ıntel'in", 1)]
```

```
print("LM Sorted Bigrams with Frequencies (first 100): \n",lm.bigrams())
```

➔ LM Sorted Bigrams with Frequencies (first 100) :

```
 [(('.', '</s>'), 19), (('<s>', 'bu'), 3), (('bilgisayar', ','), 3), (('bir',
'bilgisayar'), 3), (('<s>', '['), 2), (('<s>', 'bilgisayar'), 2), (('görevleri',
'yerine'), 2), (('"', 'işletim'), 1), (('"', 'tam'), 1), (('(', 'ana'), 1), ((')',
've'), 1), ((',', 'bellek'), 1), ((',', 'bilgi'), 1), ((',', 'bilgisayarların'), 1),
((',', 'bir'), 1), ((',', 'birbirine'), 1), ((',', 'cep'), 1), ((',', 'dizüstü'), 1),
((',', 'günümüz'), 1), ((',', 'işletim'), 1), ((',', 'karmaşıklıkları'), 1), ((',',
'kendisine'), 1), ((',', 'program'), 1), ((',', 'tarih'), 1), ((',', 'çok'), 1), ((',',
'özellikle'), 1), (('0', 've'), 1), (('1', ']'), 1), (("1'den", 'oluşan'), 1),
(('1969', 'yılında'), 1), (('1969', 'yılından'), 1), (('2', ']'), 1), (('20.',
'yüzyılın'), 1), (('21.', 'yüzyılın'), 1), (("4004'ten", 'sonra'), 1), (('<s>', '"'),
1), (('<s>', '20.'), 1), (('<s>', '21.'), 1), (('<s>', 'bilgisayarlar'), 1), (('<s>',
'bilgisayarın'), 1), (('<s>', 'church-turing'), 1), (('<s>', 'dolayısıyla'), 1),
(('<s>', 'günümüzde'), 1), (('<s>', 'ilk'), 1), (('<s>', 'istenilen'), 1), (('<s>',
'toplumumuz'), 1), (('<s>', "ıntel'in"), 1), (('[', '1'), 1), (('[', '2'), 1), ((']',
'bilgisayarın'), 1), ((']', 'günümüz'), 1), (('adlandırılabilir', '.'), 1), (('adı',
'verilen'), 1), (('alanlara', 'sığdırılabilen'), 1), (('altını', 'çizer'), 1), (('ana',
'yazılım'), 1), (('ana', 'özellikleridir'), 1), (('anlamlısı', 'ise'), 1),
(('aritmetik', 'ya'), 1), (('aydın', 'köksal'), 1), (('aynı', 'görevleri'), 1),
(('ayıran', 'ana'), 1), (('ağı', 'veya'), 1), (('bağlı', 've'), 1), (('başına',
'varıldığında'), 1), (('belirtilmektedir', '.'), 1), (('bellek', 've'), 1), (('beri',
'türkçede'), 1), (('bilgi', 'çağının'), 1), (('bilgisayar', 'ağı'), 1), (('bilgisayar',
"eniac'tır"), 1), (('bilgisayar', 'kavramıyla'), 1), (('bilgisayar', 'kümesi'), 1),
(('bilgisayar', 'sistemi'), 1), (('bilgisayar', 'sözcüğünün'), 1), (('bilgisayar',
'teknolojisi'), 1), (('bilgisayarlar', ','), 1), (('bilgisayarlar', 'bir'), 1),
(('bilgisayarlar', 'büyük'), 1), (('bilgisayarlara', 'kadar'), 1), (('bilgisayarları',
','), 1), (('bilgisayarları', 'çok'), 1), (('bilgisayarların', 'çok'), 1),
(('bilgisayarlarından', 'yüzlerce'), 1), (('bilgisayarı', 've'), 1), (('bilgisayarın',
'bir'), 1), (('bilgisayarın', 'eş'), 1), (('bilgisayarın', 'görevlerini'), 1),
(('bilgisayarın', 'temel'), 1), (('bilgisayarından', 'süper'), 1), (('bilgisayarını',
','), 1), (('bir', 'bilgisayarın'), 1), (('bir', 'dizi'), 1), (('bir', 'diğer'), 1),
(('bir', 'grup'), 1), (('bir', 'kol'), 1), (('bir', 'makinedir'), 1), (('bir', 'oda'),
1), (('bir', 'pil'), 1), (('birbirine', 'bağlı'), 1)]
```

```
lm.unigramCount('bir')            ➔ 11
lm.unigramCount('bilgisayar')     ➔ 10

lm.unigramProb('bir')             ➔ 0.031073446327683617
lm.unigramProb('bilgisayar')      ➔ 0.02824858757062147


lm.bigramCount(('bir','bilgisayar'))      ➔ 3
lm.bigramCount(('bilgisayar','bir'))      ➔ 0


lm.bigramProb(('bir','bilgisayar'))       ➔ 0.2727272727272727
lm.bigramProb(('bilgisayar','bir'))       ➔ 0
```

```
# Smoothed probabilities
lm.unigramProb_SmoothingUNK('bir')              ➔ 0.020942408376963352
lm.unigramProb_SmoothingUNK('bilgisayar')       ➔ 0.019197207678883072

lm.bigramProb_SmoothingUNK(('bir','bilgisayar'))        ➔ 0.017391304347826087
lm.bigramProb_SmoothingUNK(('bilgisayar','bir'))        ➔ 0.004366812227074236



# Sentence Probabilities
lm.sentenceProb(['<s>', 'bilgisayar', 'bir', 'dizi', 'mantık', 'işlemini', 'otomatik',
'olarak', 'yapabilen', 'bir', 'makinedir', '.', '</s>'])
➔ 8.893180106463867e-25



# Generating Sentences

lm.generateSentence()      # Most Probable Sentence
➔ ['<s>','bu','kadar',',','bellek','ve',"1'den",'oluşan','kodlamalardır','.','</s>']

# Randomly generated sentences (different sentences can be generated)
lm.generateSentence(["<s>"],2,20)
➔ ['<s>', '[', '2', ']', 'bilgisayarın', 'bir', 'bilgisayar', ',', 'bellek', 've',
"1'den", 'oluşan', 'kodlamalardır', '.', '</s>']
lm.generateSentence(["<s>"],2,20)
➔ ['<s>', 'bu', 'programlar', ',', 'bilgi', 'çağının', 'simgeleri', 'olarak',
'adlandırılabilir', '.', '</s>']
lm.generateSentence(["<s>"],3,20)
➔ ['<s>', 'bilgisayar', ',', 'bilgisayarların', 'çok', 'yaygın',
'kullanılmaktadırlar', '.', '</s>']
lm.generateSentence(["<s>"],3,20)
➔ ['<s>', 'bilgisayar', ',', 'bilgisayarların', 'çok', 'küçük', 'alanlara',
'sığdırılabilen', 'devreler', 'yapılabilmesidir', '.', '</s>']
```

Hand in:

- You will submit your programming assignment using the HADI system. You have to upload a single zip file (.zip, gzip or .rar file) holding a single python program (.py file).
- The name of your python file should be **hw02-*NameLastname*.py** by replacing ***NameLastname*** with your actual first name and your last name. Similarly, the name of your zip file name should be the same name with .zip (.gzip or .rar) extension.
- Your programming assignments will be tested the training corpora (including `hw02_tinyTestCorpus.txt`, `hw02_tinyCorpus.txt`, `hw02_bilgisayar.txt` files) as above examples. But, your assignments can be also tested with other training corpora (different files) which are not given here.
- You can use `hw02_template.py` file as a starting reference or write your own code from scratch.

Late Policy:

- You must submit your programming assignment before its due date.
- You may submit your assignment up to three days late, but with a penalty. A 10% penalty will be applied for each day late (Penalties: 1 day late: 10%, 2 days late: 20%, 3 days late: 30%)

DO YOUR PROGRAMMING ASSIGNMENTS YOURSELF!

- **Do not share your programming assignments with your friends.**
- **Do not use AI-tools (such as chatgpt) to do your programming assignments.**
- **Cheating will be punished.**