NLP - ASSIGNMENT 1

1. N-gram language model (6%)

1.1 Create n-grams for n=1, 2, 3, 4. You can show sample prints.

For this task, I implemented a Python code that reads a text corpus written in Amharic from a file. And then I tokenized the text into words and generated n-grams for n=1, 2, 3, and 4 using a sliding window approach. The generate_ngrams function takes a list of words and an integer n as input and returns a list of n-grams.

```
1-grams:
ምን
መሰላችሁ
እንባቢያን
ኢትዮጵያ
በተደጋጋሚ
ጥሪው
ደርሷት
ልትታደመው
ያልቻለችው
የአለም

2-grams:
ምን መሰላችሁ
መሰላችሁ እንባቢያን
እንባቢያን ኢትዮጵያ
ኢትዮጵያ በተደጋጋሚ
በተደጋጋሚ ጥሪው
ጥሪው ደርሷት
ደርሷት ልትታደመው
ልትታደመው ያልቻለችው
ያልቻለችው የአለም
የአለም የእግር
```

```
3-grams:
ምን መሰላችሁ እንባቢያን
መሰላችሁ እንባቢያን ኢትዮጵያ
እንባቢያን ኢትዮጵያ በተደጋጋሚ
ኢትዮጵያ በተደጋጋሚ ጥሪው
በተደጋጋሚ ጥሪው ደርሷት
ጥሪው ደርሷት ልትታደመው
ደርሷት ልትታደመው ያልቻለችው
ልትታደመው ያልቻለችው የአለም
ያልቻለችው የአለም የእግር
የአለም የእግር ኳስ

4-grams:
ምን መሰላችሁ እንባቢያን ኢትዮጵያ
መሰላችሁ እንባቢያን ኢትዮጵያ በተደጋጋሚ
እንባቢያን ኢትዮጵያ በተደጋጋሚ ጥሪው
ኢትዮጵያ በተደጋጋሚ ጥሪው ደርሷት
በተደጋጋሚ ጥሪው ደርሷት ልትታደመው
ጥሪው ደርሷት ልትታደመው ያልቻለችው
ደርሷት ልትታደመው ያልቻለችው የአለም
ልትታደመው ያልቻለችው የአለም የእግር
ያልቻለችው የአለም የእግር ኳስ
የአለም የእግር ኳስ ዋ
```

The output of the implementation looks like the above Screenshot which contains the generated output for each n grams model.

1.2 Calculate probabilities of n-grams:

For this question to calculate the probabilities of n-grams based on their occurrences in the corpus. I first declared the calculate_ngram_probabilities function which reads the corpus, tokenizes it, generates n-grams, and calculates their probabilities. I then sorted the n-grams by probability in descending order and printed the top 10 most likely n-grams for each n.

```
Top 10 most likely 1-grams:
ላይ: 0.009904
ነው: 0.009854
ውስጥ: 0.004538
ወደ: 0.004348
እና: 0.004196
ጋር: 0.003833
ግን: 0.003509
ጊዜ: 0.003186
ነገር: 0.002827
ደግሞ: 0.002708

Top 10 most likely 2-grams:
ዓ ም: 0.001237
ነገር ግን: 0.000563
ቀን ዓ: 0.000445
አዲስ አበባ: 0.000396
ብቻ ሳይሆን: 0.000385
ምክር ቤት: 0.000309
በአዲስ አበባ: 0.000302
ይሁን እንጂ: 0.000246
የአዲስ አበባ: 0.000244
ጠቅላይ ሚኒስትር: 0.000244
```

```
Top 10 most likely 3-grams:
ቀን ዓ ም: 0.000435
እ ኤ አ: 0.000207
ዓ ም ጀምሮ: 0.000078
ተወካዮች ምክር ቤት: 0.000073
በሌላ በኩል ደግሞ: 0.000065
በ ዓ ም: 0.000064
የአዲስ አበባ ከተማ: 0.000058
በዓለም አቀፍ ደረጃ: 0.000054
ከጊዜ ወደ ጊዜ: 0.000052
ዓ/ም ኢሳት ዜና: 0.000052

Top 10 most likely 4-grams:
ዓ ም ኢሳት ዜና: 0.000048
መጋቢት ቀን ዓ ም: 0.000034
የካቲት ቀን ዓ ም: 0.000033
ግንቦት ቀን ዓ ም: 0.000033
ሰኔ ቀን ዓ ም: 0.000030
የአዲስ አበባ ከተማ አስተዳደር: 0.000029
ጥር ቀን ዓ ም: 0.000029
ጥቅምት ቀን ዓ ም: 0.000029
በ የኔ ሃሳብ ዓምድ: 0.000027
ቀን ዓ ም ጀምሮ: 0.000027
```

The above Screen shot is the output of the python file which displays the top-most likely n-grams produced by the model.

1.3 Probability of a given sentence:

To work on this question first I initiated a function calculate_sentence_probability to calculate the probability of a given sentence. The function reads the corpus, tokenizes it, and counts the occurrences of the given sentence. The probability is calculated as the count of the target sentence divided by the total number of sentences in the corpus.

```
Probability of sentence "ኢትዮጵያ ታሪካዊ ሀገር ናት": 1.000000
```

```
print("A")
```

The probability of the sentence was 1.000000.

1.4 Generate random sentences using n-grams:

For this question first I created a function generate_random_sentence to generate random sentences using n-grams. The function starts with a randomly chosen starting word and generates the rest of the sentence based on the n-grams. I changed the n values so that i can observe the change in the output.I increased the value of n and i saw how much the output was giving a meaning that the less n value.

```
የኢትዮጵያ ሕዝብ
ነገር ግን
ብቻ ነው ብሎ
የውጭ ጉዳይ ነው
በዚህ ጊዜ ውስጥ
```

+ Code + Mar

```
ዋና ሥራ አስኪያጅ አቶ
የእድገትና ትራንስፎርሜሽን እቅድ
የካቲት ቀን ዓ ም ጀምሮ
የአየር ንብረት ለውጥ
ነው ነገር ግን
```

+ Code + Markdow

```
ባሕረ ሐሳብ የዘመን መቁጠሪያ
የኢትዮጵያ ኦርቶዶክስ ተዋሕዶ ቤተ ክርስቲያን
ግንቦት ቀን ዓ ም ድረስ
ነጻ ሃሳብ የሚስተናገድ ሲሆን ከአንባቢያን የሚላኩልን ጽሁፎች
መመሪያ መሠረት ያለ መድልዎ በዚህ ዓምድ ሥር
```

+ Code + Markdown

In the above Screenshot you can observe how much difference it has when we are increasing the value of n. As n increases the message that was to be transferred from the sentence makes more meaning than the one with less n value.

2 Evaluate these Language Models Using Intrinsic Evaluation Method (4%)

While trying to evaluate my ngram model using intrinsic method i followed This steps:-

- *remove_symbols: Removes non-Amharic-alphabetic characters and symbols from a word.*
- *clean_text: Applies remove_symbols to each word in a line.*
- *ngrams_model: Takes a line of text and generates n-grams. It uses a sliding window approach to create tuples of words for each n-gram.*
- *calculate_perplexity: Takes an n-gram model, a validation set, and the value of 'n'. It calculates the log probability for each n-gram in the validation set and sums up these log probabilities. The perplexity is then calculated using the formula.*
- *The code loads the corpus, shuffles it, and splits it into training and validation sets.*
- *It uses a Counter to store the counts of n-grams in the training set.*
- *It calculates perplexity on the validation set using the n-gram model.*

#3 Evaluate these Language Models Using Extrinsic Evaluation Method (4%)

Here's a step-by-step description of the What we came up with:

Data Cleaning:

*The **remove_symbols** function removes non-Amharic alphabetic characters and symbols from a given word.*

*The **clean_text** function tokenizes a line of text into words and then applies the remove_symbols function to each word.*

Data Splitting:

*The **split_data** function takes a corpus and a split ratio as input and shuffles the data. It then splits the data into training and validation sets based on the specified split ratio.*

n-gram Model Creation:

The ngrams_model function generates n-grams from the cleaned text data. It splits each line into words and adds the words as n-grams to the list.

Reading Amharic Corpus:

The Amharic corpus is read from a file specified by corpus_file_path.

Training n-gram Model:

The code uses the split_data function to split the corpus into training and validation sets. It then uses the ngrams_model function to create a n-gram model from the training set.

Printing Debug Information:

The code prints the number of n-grams in the training set and displays a sample of n-grams.

Extrinsic Evaluation:

It samples a random line from the validation set and generates text for evaluation using the n-gram model. The generated text is then printed for inspection.

Extrinsic Evaluation:

The extrinsic evaluation here involves generating text using the n-gram model and comparing it with a randomly selected line from the validation set. The goal is to see how well the model can generate text that resembles the language used in the validation set.

THANK YOU!