Language Modelling

Assignment Details

- **Motivation**: The motivation of this assignment is to gain insight into language modelling.
- **Problem Statement**: The goal of the assignment is to build N-gram language models and implement a next-word prediction program.
- **Assignment Parts**:
    - Part 1: Build N-gram models (bi-gram, tri-gram, four-gram, five-gram) from a provided dataset.
    - Part 2: Implement a program for predicting the next word based on the N-gram models.

Part 1: Building N-gram Models

In this part of the assignment, i mplemented a Python program to build N-gram language models. The following steps were followed:

1. **Data Input**: obtained a dataset (in our case, "input_dataset.txt") containing a list of sentences, with one sentence per line.
2. **Tokenization**: Each sentence in the dataset was tokenized into words using Python's string split() function.
3. **N-gram Creation**: N-grams (bi-grams, tri-grams, four-grams, and five-grams) were generated from the tokenized words.
4. **Frequency Count**: counted the frequency of each N-gram in the dataset.
5. **Output Format**: The N-gram models were written to separate output files following the specified format, where each N-gram was accompanied by its count.

Part 2: Predicting the Next Word

In the second part of the assignment, implemented a program for predicting the next word given a set of input words using the N-gram models generated in Part 1. The program followed these steps:

1. **N-gram Model Loading**: loaded the appropriate N-gram models (2-grams in our example) from the output file.
2. **User Interaction**: The program interacted with the user by continuously prompting for a set of input words.
3. **Prediction**: The program predicted the next word based on the input words and the N-gram models using the `predict_next_word` function.
4. **Output**: The predicted next word was displayed to the user.

Usage Instructions

To run the program, execute the Python script "MIS-No_LM_Part2.py" with Python 3. You can enter a set of words as input to receive predictions for the next word.

bash

```
python3 MIS-No_LM_Part2.py
```

Sample Outputs

- Sample input: "my name"
  - Predicted Next Word: "No prediction available."
- Sample input: (no input)
  - The program waits for user input.

Conclusion

In conclusion, this assignment allowed us to gain hands-on experience in language modelling, particularly in building N-gram models and using them for next-word prediction. successfully implemented the required functions and programs to achieve the assignment's objectives.