**N-grams: Summary and Insights**

**Objective**

This project aimed to examine how well a statistical language model based on n-grams can predict the next word in a sequence. Using the Brown corpus, we applied basic frequency-based techniques to forecast upcoming words from a given phrase.

**Corpus & Preprocessing**

We worked with the Brown corpus from NLTK, known for its balanced mix of text types, making it suitable for language modeling.  
Our preprocessing steps included:

* Converting all words to lowercase
* Filtering out punctuation and numeric values by checking if each word is alphabetic (word.isalpha())

These steps ensured a clean set of meaningful language tokens for analysis.

**Methodology**

1. **Building Trigrams**  
   We generated trigrams (n=3) using nltk.util.ngrams. These were then counted and stored in a frequency-sorted DataFrame.

**Why Trigrams?**  
Trigrams offer a middle ground: they capture more context than bigrams but are less sparse than higher-order n-grams.

**Examples of Frequent Trigrams:**

* 1. **('one', 'of', 'the') – 404 times**
     + Common phrase structure in English
     + Found in expressions like "one of the best"
  2. **('the', 'united', 'states') – 337 times**
     + A frequent named entity in political or historical contexts
     + Reflects the American focus of the Brown corpus
  3. **('as', 'well', 'as') – 238 times**
     + Widely used comparative phrase
     + Functions like a conjunction
  4. **('some', 'of', 'the') – 179 times**
     + Grammatical pattern leading into noun phrases
     + Indicates subsets or partial amounts
  5. **('out', 'of', 'the') – 174 times**
     + Common in idioms or prepositional phrases like "out of the question"

1. **Next-Word Prediction Function**  
   We created a function that identifies the most frequent next words based on trigram matches for a given phrase.

Key features:

* 1. Checks input length (n-1 tokens)
  2. Matches prefixes with trigrams
  3. Returns top-k next-word suggestions by frequency

**Example Output:**  
(Example not shown in original but implied)

**Interpretation and Key Takeaways**

| **Metric** | **Notes** |
| --- | --- |
| **Relevance** | Predictions are valid and grammatically correct |
| **Context Awareness** | Limited—no semantic understanding, only word co-occurrence |
| **Bias** | Reflects frequency and themes of the Brown corpus (1950s–60s content) |
| **Performance** | Fast and simple, but limited in broader language understanding |

**Strengths:**

* Very fast and lightweight
* Easy to interpret and explain
* Suitable as a starting point in low-resource NLP settings

**Weaknesses:**

* Can't handle unseen input sequences
* Based solely on word frequency, not meaning
* Lacks smoothing or backoff strategies

**Conclusion**

This project showed that n-gram models, despite their simplicity, can produce reasonable next-word predictions using frequency data. While they lack deep contextual understanding, they are useful in early-stage NLP projects or as baselines for evaluating more advanced models.