**Difference between N-gram and RNN**

**1. Definition:**

* **N-gram:** An N-gram is a sequence of 'n' contiguous words or characters from a given text. It is commonly used in probabilistic models for predicting the next word in a sentence based on the previous words. N-grams are considered a traditional approach for sequence prediction in natural language processing (NLP).
* **RNN (Recurrent Neural Network):** RNN is a type of artificial neural network designed to recognize patterns in sequences of data, such as text, time series, or speech. It maintains a hidden state that captures information about previous inputs, allowing it to model sequential data over time.

**2. Context Handling:**

* **N-gram:** N-grams have a fixed window size (the "n" in N-gram) and only consider a limited number of previous words (n-1) to predict the next word. They struggle with long-term dependencies since the context is restricted to a fixed length.
* **RNN:** RNNs can handle long-term dependencies better as they maintain a hidden state that is updated at each time step, allowing information from previous words (or time steps) to influence future predictions. This makes RNNs more powerful for capturing long-term patterns.

**3. Model Complexity:**

* **N-gram:** The model is relatively simple and computationally efficient, but it becomes less effective as "n" increases due to the curse of dimensionality (the number of possible sequences grows exponentially with "n").
* **RNN:** RNNs are more complex, involving many layers and parameters, but they are more expressive and capable of learning complex relationships in the data. However, they require more computational resources and training time.

**4. Ability to Generalize:**

* **N-gram:** N-grams can fail to generalize well to unseen sequences, especially when the training data is sparse, since they rely on observing specific n-word combinations.
* **RNN:** RNNs can generalize better to unseen data because they learn representations of sequences instead of relying on explicit memorization of word combinations.

**Advantages of RNN over N-gram:**

1. **Capturing Long-Term Dependencies:**
   * RNNs are better at capturing relationships between words that are far apart in a sequence, whereas N-grams can only look at a fixed, short window of context.
2. **Handling Variable-Length Sequences:**
   * RNNs can process input sequences of varying lengths, whereas N-grams require a fixed "n" size, limiting their ability to adapt to different types of sequences.
3. **Better Generalization:**
   * RNNs learn distributed representations (i.e., word embeddings), which enable better generalization and the ability to handle unseen sequences more effectively than N-grams.
4. **Dynamic Context Adaptation:**
   * RNNs maintain an evolving hidden state, which adapts as new input is processed, whereas N-grams have a static context size and are unable to adjust to new information dynamically.
5. **Handling Sequential Data Beyond Text:**
   * While N-grams are mainly used for textual data, RNNs can handle a variety of sequential data, such as time series and speech recognition tasks, making them more versatile.