**Text Generation with Transformers: A Diagrammatic Approach**

Here's a breakdown of text generation with transformers using a diagram:

**1. Input & Preprocessing:**

* **Seed Text:** You provide a starting sentence or phrase (e.g., "Once upon a time").
* **Tokenizer:** This component converts the text into a sequence of numerical tokens the model understands (e.g., "Once" -> 1234, "upon" -> 5678).

**2. Transformer Model:**

* **Encoder:**
  + Processes the tokenized seed text (e.g., 1234, 5678).
  + Uses self-attention to understand relationships between words based on their positions and context.
  + Generates a contextual representation for each token.
* **Decoder:**
  + Takes the encoded representation from the encoder.
  + Uses attention again, but this time it focuses on both the encoded representation and the previously generated text.
  + Predicts the probability distribution for the next word in the sequence.

**3. Text Generation Loop:**

* **Prediction:**
  + The decoder samples a word from the predicted probability distribution. This sampling can be influenced by a "temperature" parameter that controls randomness.
* **Output & Update:**
  + The sampled word is converted back to text (e.g., 9012 -> "a").
  + The generated word is appended to the existing seed text, forming a new sequence for the next prediction step.
* **Loop Termination:**
  + The loop continues until a specific end-of-sequence (EOS) token is predicted (e.g., "<EOS>") or a maximum length is reached.

**4. Output:**

* The final generated text is returned, which is a continuation of the seed text based on the model's understanding of language patterns.

