### RESPUESTAS TEORICAS.

### Diferencias entre los modelos encoder-only, decoder-only y encoder-decoder en el contexto de los chatbots conversacionales:

**Modelos Encoder-Only:**

* **Descripción**: Un modelo encoder-only, como **BERT** (Bidirectional Encoder Representations from Transformers), se utiliza principalmente para tareas de clasificación o etiquetado de secuencias. El encoder procesa la entrada y genera representaciones contextuales de las palabras sin intentar generar un texto de salida. En el contexto de chatbots, este tipo de modelo no es adecuado para generar respuestas, ya que está diseñado para comprender y extraer características de una secuencia de entrada.
* **Usos típicos**: Estos modelos se usan principalmente en tareas de clasificación de texto, análisis de sentimientos, extracción de entidades o para tareas que impliquen análisis de la entrada sin necesidad de generar texto de salida.

**Modelos Decoder-Only:**

* **Descripción**: Los modelos decoder-only, como **GPT-2** y **GPT-3**, están diseñados específicamente para tareas de generación de texto. Estos modelos toman una secuencia de entrada (como un prompt) y, a través de un mecanismo de atención autoregresiva, generan texto secuencialmente, prediciendo la siguiente palabra a partir del contexto anterior. En el contexto de los chatbots, los modelos decoder-only son altamente efectivos porque pueden generar respuestas coherentes de manera autónoma a partir de un mensaje de entrada.
* **Usos típicos**: Este tipo de modelo es excelente para tareas de generación de texto libre, como responder preguntas, completar frases, o generar diálogos completos.

**Modelos Encoder-Decoder:**

* **Descripción**: Los modelos encoder-decoder, como **T5** y **BART**, combinan un encoder y un decoder para transformar una entrada en una salida. El encoder convierte la entrada en una representación interna que captura su contexto, y el decoder utiliza esta representación para generar una salida. Estos modelos son muy eficaces cuando se necesita comprender una entrada de manera profunda y luego generar una respuesta adecuada, lo que es esencial en tareas de traducción automática y resumen de texto.
* **Usos típicos**: Los modelos encoder-decoder son ideales para tareas donde el contexto completo de la entrada debe ser procesado para generar una respuesta relevante, como en la traducción de idiomas, generación de texto a partir de preguntas complejas, y chatbots conversacionales que requieren un análisis más profundo de la entrada.

**Decisión de modelo más adecuado:**

* **Encoder-Only**: Mejor para tareas de comprensión (clasificación, análisis de sentimientos, etc.), pero no es adecuado para la generación de respuestas.
* **Decoder-Only**: Ideal para chatbots conversacionales simples que generan respuestas a partir de un prompt. Es el modelo más adecuado para generar respuestas de texto autónomamente.
* **Encoder-Decoder**: El más adecuado para chatbots que requieren generar respuestas complejas basadas en una comprensión profunda del contexto, como responder preguntas complejas o proporcionar resúmenes.

**Concepto de "temperatura" en la generación de texto con LLMs y su impacto en el comportamiento del chatbot:**

**Definición de temperatura:**

* **Temperatura** es un parámetro utilizado en los modelos de lenguaje generativos que controla la aleatoriedad en la generación de texto. En términos sencillos, regula cómo el modelo elige entre las posibles siguientes palabras durante la generación de texto.
  + **Temperatura baja** (cerca de 0) hace que el modelo sea más determinista. El modelo seleccionará las palabras más probables, resultando en respuestas más coherentes y predecibles.
  + **Temperatura alta** (por ejemplo, 1 o más) aumenta la aleatoriedad, lo que permite que el modelo elija palabras menos probables y produzca respuestas más variadas y creativas, pero a costa de una mayor probabilidad de incoherencia.

**Impacto de la temperatura en el chatbot:**

* **Temperatura baja (ej., 0.2 - 0.5)**: Cuando la temperatura es baja, el modelo tiende a generar respuestas más coherentes y seguras, lo cual es beneficioso cuando se quiere que el chatbot sea preciso y confiable, especialmente en aplicaciones como asistentes virtuales, soporte técnico, o chatbots informativos. Sin embargo, esto puede llevar a respuestas monótonas o menos creativas.
* **Temperatura alta (ej., 0.7 - 1.0)**: Una temperatura más alta genera respuestas más creativas y diversas, lo que puede ser útil en aplicaciones donde se requiere creatividad, como en la escritura de historias, generación de diálogos complejos o juegos interactivos. Sin embargo, a temperaturas más altas, el modelo tiene más probabilidades de generar respuestas irrelevantes o erróneas.

**Consideraciones al ajustar la temperatura:**

* **Aplicaciones que requieren precisión**: Si la tarea es proporcionar respuestas exactas, como en el soporte técnico o un chatbot educativo, es mejor usar una temperatura baja para garantizar respuestas coherentes.
* **Aplicaciones creativas o de entretenimiento**: Para tareas que requieren generación creativa de contenido, como juegos o narración de historias, una temperatura más alta puede ser adecuada para generar respuestas diversas y sorprendentes.

**Técnicas para reducir las "alucinaciones" en chatbots basados en LLMs: Alucinaciones en LLMs:**

Las alucinaciones en los modelos de lenguaje se refieren a respuestas que parecen razonables o coherentes pero que son factualmente incorrectas o inventadas. Este es un problema común con los modelos generativos, ya que pueden crear información plausible pero incorrecta, especialmente cuando se les pide que generen respuestas a partir de información que no tienen en su conjunto de entrenamiento.

**Técnicas para reducir las alucinaciones:**

**A nivel de inferencia:**

1. **Uso de modelos de verificación de hechos**: Combina el modelo generativo con un modelo adicional de verificación de hechos. El modelo de verificación podría comprobar la veracidad de las afirmaciones generadas antes de presentarlas al usuario.
   * Ejemplo: Consultar una base de datos o un motor de búsqueda en tiempo real (como **Wikipedia** o APIs de verificación de hechos) para confirmar la precisión de la respuesta antes de entregarla.
2. **Control de la temperatura**: Reducir la temperatura en el modelo generativo puede disminuir la probabilidad de que el modelo "alucine". Con temperaturas bajas, el modelo tiende a elegir las respuestas más probables y coherentes, lo que reduce la probabilidad de generar información errónea.
3. **Uso de respuestas con probabilidades más altas**: Si bien la temperatura controla la aleatoriedad, también se pueden filtrar las respuestas generadas para aceptar solo aquellas con alta probabilidad. Esto limita las respuestas generadas a las más seguras y precisas

* **A nivel de "prompt engineering":**

1. **Instrucciones explícitas**: Diseñar los prompts para que sean más precisos y guíen al modelo a generar respuestas correctas. Por ejemplo, proporcionar explícitamente en el prompt que la respuesta debe basarse en hechos verificables o que se debe limitar a conocimientos generales.
   * Ejemplo: "Por favor, responde solo con hechos verificables hasta 2021, sin inventar información."
2. **Respuestas formateadas**: Utilizar prompts que estructuren las respuestas de manera que proporcionen detalles claros y específicos, reduciendo la probabilidad de que el modelo haga suposiciones incorrectas o genere respuestas vagas.
3. **Técnicas de fine-tuning**: Entrenar el modelo en un conjunto de datos más específico y curado puede ayudar a reducir las alucinaciones. Al mejorar la calidad del conjunto de entrenamiento, el modelo aprenderá a generar respuestas más precisas.