Estrategias de decodificación

Ignacio Pedrero Muro

9 de octubre de 2023

1. Introducción

La generación de texto es una parte esencial de los LLM, pero las estrategias de decodificación que juegan un papel crucial en este proceso a menudo se pasan por alto. En este proyecto, se explorará en detalle el mecanismo detrás de la generación de texto en LLMs y se discutirán técnicas como la búsqueda codiciosa y la búsqueda de haz.
En esta tarea trataremos de comprender y analizar cómo los LLM, como GPT-2, generan texto y cómo se pueden mejorar las salidas utilizando diferentes estrategias de decodificación

2. Descripción Detallada de LLM

2.1. Explicación de cómo LLMs, como GPT-2, no producen texto directamente sino logits.

Los modelos de lenguaje de gran tamaño, como GPT-2 se emplean para la creacion de texto (entre otras cosas) con el fin de aproximarse al lenguaje natural. Para ello se les entrena con cantidades gigantes de datos, de los cuales estos programas tratan de identificar patrones entre los datos con el fin de hacer predicciones. Al analizar estos patrones se generan grafos donde cada palabra es un vertice que tiene conexion con los otros vertices mediante aristas. Estas aristas a su vez contienen la probabilidad (segun los datos del entrenamiento) de que un vertice pase a otro vertice, y eso es lo que se llama logit. Un logit es un valor numerico que indica la probabilidad que tiene un vertice de saltar a otro vertice, es decir de que una palabra venga despues de otra palabra ya conocida. Los LLMs se dedican a recaudar la informacion de los logits y los vertices, y con esa informacion se emplean diferentes algoritmos que generan palabras en funcion de la probabilidad según su modelo de aprendizaje

2.2. Análisis de cómo los logits se traducen en texto.

Una vez tenemos un texto de entrada, debemos de buscar los logits con mayor probabilidad, ya que estos son los que mas se emplean en nuestro lenguaje. Hay varios algortimos de generación de texto, pero todos se basan en analizar el logit con mayor probabilidad despues del ultimo vertice, o en analizar combinaciones de vertices (palabras) y aristas (logits), con el fin de seleccionar aquellas combinaciones que son mas probables estadisticamente.

3. Proceso de Generación de Texto

3.1. Descripción del proceso, desde la entrada de texto hasta la predicción del siguiente token.

Cuando nosotros introducimos un texto en los MMLS, estos lo que hacen es guardar el token y buscar los logits mas altos en su base de datos, luego en función de que algoritmo este empleando, nos devolvera una lista de tokens (es decir, una frase en lenguaje natural) que contenga una suma alta de probabilidad en su modelo de aprendizaje, es decir busca las palabras que mas aparecen despues de una palabra con el fin de devolver una palabra u oración coherente. Hay algoritmos que cogen el logit mas alto, otros analizan los siguientes -n tokens y hacen una media normalizada de los logits para ver que esa oración tenga sentido y otros buscan en los logits mas usados y devuelven con distintas

probabilidades diferentes valores, haciendolo asi mas aleatorio pero de manera que tenga sentido en nuestro lenguaje.

3.2. Representación ilustrativa de la conversión de token a ID de token y viceversa.

Los ID de tokens son la representación numerica de los tokens, es decir, a cada token se le asigna un valor numérico, lo que hace mas facil que el modelo de lenguaje los identifique y obtenga sus logits mediante operaciones matematicas con esos ID, ademas es mas facil para la máquina analizar asi sus patrones.

4. Estrategias de Decodificación

4.1. Descripción y análisis de la búsqueda codiciosa y la búsqueda de haz.

Busqueda codiciosa:
consiste en un método de decodificación que toma el token más probable en cada paso como el siguiente token en la secuencia. Es decir solo busca el logit con el valor mas alto y nos devuelve el token correspondiente sin mirar el resto de tokens.
Busqueda de haz:
Esta funcion tiene en cuenta los -n tokens mas probables y calcula las puntuaciones de los tokens (normalizadas, para que la longitud no influya) con el fin de hallar la secuencia con un valor mas alto y luego buscara el camino mas corto desde el nodo base hasta el nodo final que cierra la secuencia. Es decir genera varias posibilidades usando diferentes tokens para crear un grafo, todas las ramas del grafo terminan con un token de cierre. Cada rama tiene un valor en función de sus logits. Lo que hace este algoritmo es buscar la manera optima de conectar el primer nodo con el último de la secuencia de mayor valor.

4.2. Discusión sobre muestreo con top-k y muestreo de núcleo.

Muestreo top-k:
Esta técnica atribuye una probabilidad a cada token y luego se queda con la cantidad de tokens que eliga (-k), es decir por ejemplo coge los 5 mas probables. Para generar texto con ellos, lo que hace es aleatoriamente generar uno de esos tokens, cada uno con su probabilidad de aparecer, lo que provova que los tokens mas probables aparezcan mas veces a la vez que aumenta la aleatoriedad de los textos generados.
Muestreo de nucleo:
Este metodo selecciona los k tokens más probables, de modo que elige un valor de corte p tal que la suma de las probabilidades de las fichas seleccionadas exceda p. Es decir, examina sus tokens más probables en orden descendente y continúa agregándolos a la lista hasta que la probabilidad total supera el umbral p. Una vez supera ese umbral genera (con los tokens seleccionados) una respuesta, empleando cada token con su probabilidad correspondiente, de manera que en cada paso selecciona varios tokens aspirantes y luego los genera aleatoriamente dando asi respuestas mas variadas y originales.

5. Hiperparámetros y su Manipulación

5.1. Discusión sobre temperatura, num beams, top k y top p y ejemplos de cómo manipular estos hiperparámetros puede afectar la salida.

Temperatura:
Se utiliza para controlar el grado de aleatoriedad en las predicciones de palabras que hace el modelo durante la generación de texto. Ajustar la temperatura puede influir en la coherencia y diversidad del texto generado, por ejemplo una temperatura alta hace que las predicciones del modelo sean más aleatorias. Esto significa que el modelo considera una amplia variedad de palabras como posibles siguientes tokens y tiene menos restricciones en la elección de palabras por lo que tiende a ser más diverso y creativo, pero a veces puede ser menos coherente o relevante para el contexto. Una temperatura baja

implica que las predicciones del modelo sean más determinístas El modelo tiende a seleccionar las palabras más probables en función del contexto, lo que resulta en un texto más coherente y predecible.
Numero de beams:

Con un solo beam, el modelo elegira la palabra siguiente en función de la probabilidad más alta en cada paso de generación. Esto tiende a generar un texto más determinístico, ya que solo se considera una única opción en cada paso.
Sin embargo cuando se utiliza un número mayor de beams el modelo considera varias opciones de palabras siguientes en cada paso lo que permite explorar diferentes caminos en la generación de texto y, a menudo, produce resultados más diversos y creativos. Sin embargo, aumentar el número de beams también aumenta la complejidad computacional y puede ralentizar la generación de texto.
Top K:
Un valor pequeño de top k limitará el conjunto de palabras consideradas a solo las 10 palabras más probables, lo que tiende a generar texto más coherente pero posiblemente menos creativo. Sin embargo un valor grande de top k como 50 o más permite que el modelo considere una gama más amplia de palabras como opciones, lo que puede aumentar la diversidad y la creatividad del texto generado, pero a veces a costa de la coherencia y de velocidad de procesamiento.
Top p:
Un valor más alto de "top p"permite que se consideren más palabras, lo que puede aumentar la diversidad, mientras que un valor más bajo limita las opciones y favorece la coherencia.

6. Reflexión y Conclusiones

6.1. Reflexiones sobre el impacto y la importancia de las estrategias de decodificación en la generación de texto.

Las estrategias de decodificación influyen en la coherencia y la relevancia del texto generado, estas estrategias ayudan a garantizar que el texto sea coherente con el contexto previo y sea relevante para la tarea o el propósito específico.

Las estrategias de decodificación brindan a los usuarios un mayor control sobre la generación de texto. Los parámetros como la temperatura, "top_k","top_p", $yelnúmerodebeamspermitenajustarlasalidadelmodelosegúnlasp$

También afecta a la eficiencia computacional, ya que algunas estrategias de decodificación pueden ser más computacionalmente costosas que otras.

En resumen, las estrategias de decodificación son esenciales en la generación de texto, ya que afectan la coherencia, la diversidad, la relevancia y la creatividad del texto generado. Comprender cómo funcionan estas estrategias y cómo ajustar los parámetros asociados es crucial para obtener resultados óptimos y útiles en una variedad de aplicaciones de procesamiento de lenguaje natural, ademas puede facilitar la identificación de problemas de generación y la mejora de los modelos.

6.2. Conclusiones sobre el tema y sugerencias para futuras investigaciones.

Es muy interesante saber como generan texto los LLms y analizar las diferentes estrategias para que el texto solicitado varie segun algunos hiperparametros.
Una futura investigacion interesante podria ser sobre los diferentes algoritmos que generan distintos tipos de grafos con cantidades enormes de datos y analizar como se interpretan esos datos matematicamente para que el ordenador sea caaz de procesarlos tan rapido y como se guardan luego los resultados con el fin de emplearlos en cualquier otro contexto.