Contents

1 Revisión de Indexación de Base de Datos y Organización de Archivos 4

2 Guía de Estudio de Álgebra Relacional 6

2.1 Preguntas de Respuesta Corta 6
2.2 Clave de Respuestas Cortas 7
2.3 Glosario de Términos Clave 8
2.3.1 Relación 8
2.3.2 Esquema 8
2.3.3 Tupla 8
2.3.4 Atributo 8
2.3.5 Dominio 8
2.3.6 Álgebra Relacional 8
2.3.7 Operador Lógico 9
2.3.8 Operador Físico 9
2.3.9 Optimización de Consultas 9
2.3.10 Transferencia de Bloque de Disco 9
2.3.11 Índice B+ Tree 9
2.3.12 Factor de Selectividad 9
2.3.13 Join de Bucle Anidado 9
2.3.14 Merge-Join 10
2.3.15 Eliminación de Duplicados 10
2.3.16 Ordenamiento Externo 10
2.3.17 Concatenación (Union All) 10
2.3.18 Intersección 10
2.3.19 Diferencia (Minus) 10

3 Guía de Estudio de Álgebra Relacional y Optimización de Consultas 11

3.1 Preguntas de Respuesta Corta 11
3.2 Clave de Respuestas Cortas 11
3.3 Glosario 12
3.3.1 Query Evaluation Plan 12
3.3.2 Query Optimization 12
3.3.3 Materialization 13
3.3.4 Pipelining 13
3.3.5 Selectivity Factor 13
3.3.6 “Pushing Selections” 13
3.3.7 Join Order 13
3.3.8 Left-Deep Join Tree 13
3.3.9 Heuristic 14
3.3.10 Cost-Based Optimization 14
3.3.11 Dynamic Programming 14
3.3.12 Hybrid Query Optimization 14

4 Guía de Estudio de Recuperación de Información **14**
4.1 Preguntas de Respuesta Corta 14
4.2 Clave de Respuestas Cortas 15
4.3 Glosario de Términos Clave 16

5 Guía de Estudio de Recuperación de Información Web **17**
5.0.1 Glosario de Términos Clave 17
5.0.2 Preguntas 18
5.0.3 Respuestas Cortas 18

6 Guía de Estudio de Procesamiento de Lenguaje Natural **19**
6.1 Glosario de Términos Clave 19
6.1.1 Procesamiento de Lenguaje Natural (NLP) 19
6.1.2 Análisis Sintáctico 19
6.1.3 Análisis Semántico 20
6.1.4 Reconocimiento de Entidades Nombradas (NER) 20
6.1.5 Generación de Lenguaje Natural (NLG) 20
6.1.6 Análisis de Sentimientos 20
6.1.7 Comprensión del Lenguaje Natural (NLU) 20
6.1.8 Tokenización 20
6.1.9 Stop Words 20
6.1.10 Stemming 20
6.1.11 Lematización 21
6.1.12 Etiquetado de Partes del Discurso 21
6.1.13 Contexto 21
6.1.14 Contexto Local 21
6.1.15 Contexto Global 21
6.1.16 Contexto Conversacional 21
6.1.17 Contexto Situacional 21
6.1.18 Contexto Histórico 21
6.1.19 Contexto Cultural 22
6.1.20 Resolución de Correferencias 22
6.1.21 Análisis Pragmático 22
6.2 Cuestionario Corto 22
6.2.1 ¿Qué es el Procesamiento de Lenguaje Natural (NLP) y cuál es su objetivo principal? 22
6.2.2 Diferencie entre Análisis Sintáctico y Análisis Semántico en NLP. Proporcione un ejemplo para cada uno. 22
6.2.3 Explique la función del Reconocimiento de Entidades Nombradas (NER) con un ejemplo. 22
6.2.4 ¿Cuáles son las principales diferencias entre Comprensión del Lenguaje Natural (NLU) y Generación de Lenguaje Natural (NLG)? 23
6.2.5 Describa dos aplicaciones comunes de NLU en la vida cotidiana. . 23

6.2.6 Explique brevemente el propósito e importancia del pre-procesamiento en NLP. 23
6.2.7 ¿Qué son las "stop words" en NLP y por qué se eliminan típicamente durante el pre-procesamiento? 23
6.2.8 ¿Cuál es la diferencia entre Stemming y Lematización? Proporcione un ejemplo para cada uno. 23
6.2.9 Defina "contexto" en NLP y explique por qué es crucial para una interpretación precisa del lenguaje. 24
6.2.10 Describa dos tipos de contexto en NLP y proporcione un ejemplo de cómo cada tipo influye en la comprensión del lenguaje. 24

7 Guía de Estudio de Ingeniería de Prompts 24
7.1 Cuestionario de Respuesta Corta 24
7.1.1 1. **¿Qué es un prompt en el contexto del Procesamiento de Lenguaje Natural (NLP)?** 24
7.1.2 2. **Explique el concepto de ingeniería de prompts y su importancia en NLP.** 24
7.1.3 3. **Describa tres modificaciones específicas que caen bajo el paraguas de la ingeniería de prompts.** 25
7.1.4 4. **¿Qué es la tokenización y por qué es relevante para trabajar con modelos de lenguaje grandes?** 25
7.1.5 5. **¿Cómo afecta el parámetro "temperatura" la salida de un modelo de lenguaje en el OpenAI playground?** 25
7.1.6 6. **Explique cómo el parámetro "TopP" influye en el rango creativo y la precisión de las respuestas de un modelo de lenguaje.** 25
7.1.7 7. **Diferencie entre "Frecuencia Penalty" y "Presence Penalty" como parámetros en el OpenAI playground.** . . . 25
7.1.8 8. **Nombra y describe los tres elementos fundamentales que componen un prompt.** 26
7.1.9 9. **¿Cuál es la diferencia entre un "Standard Prompt" y un "Chain of Thought" prompt?** 26
7.1.10 10. **Explique el concepto de un "Few-Shot Prompt" y proporcione un ejemplo de su aplicación.** 26
7.2 Glosario de Términos Clave 26

8 Chatbots Conversacionales Inteligentes: Una Guía de Estudio 27
8.1 Cuestionario 27
8.1.1 1. ¿Qué son los modelos de texto y qué tareas pueden realizar? Proporcione un ejemplo de un modelo de texto bien conocido. . . . 27
8.1.2 2. Explique el concepto de modelado de lenguaje y cómo aprovecha las relaciones predecibles entre palabras. 27
8.1.3 3. Diferencie entre Modelos de Lenguaje Grandes (LLMs) y modelos de lenguaje tradicionales. 27

8.1.4 4. Describa las características clave de un modelo conversacional y proporcione un ejemplo. 28
8.1.5 5. Esboce las tres etapas principales involucradas en la creación de un modelo de IA. 28
8.1.6 6. ¿Cuáles son las consideraciones clave durante la fase de validación del desarrollo del modelo de IA? 28
8.1.7 7. Explique el propósito y los beneficios de ajustar un modelo de IA. 28
8.1.8 8. Describa el papel de un componente de gestión de diálogos en la arquitectura de un chatbot conversacional. 28
8.1.9 9. ¿Cómo difiere un modelo de lenguaje de propósito general de un chatbot conversacional inteligente en términos de gestión de diálogos? 29
8.1.10 10. ¿Cuáles son las funciones primarias de un codificador en una arquitectura de transformador? 29
8.2 Glosario de Términos Clave 29

1 Revisión de Indexación de Base de Datos y Organización de Archivos

Preguntas de Respuesta Corta:

1. **Explica cómo se organiza la data en un hard disk drive en términos de tracks, sectors y blocks.**
2. **¿Cuál es la idea principal detrás del enfoque de almacenamiento de registros de longitud fija, y qué modificación se hace para evitar accesos adicionales a bloques?**
3. **Describe dos enfoques diferentes para manejar la eliminación de registros en un archivo.**
4. **Diferencia entre la organización de archivos heap y secuencial en términos de colocación de registros.**
5. **Explica por qué la organización de archivos secuencial podría ser ineficiente para consultas que involucran el natural join de dos tablas.**
6. **¿Cuál es el propósito de un index en una base de datos, y cuáles son las compensaciones de usar uno?**
7. **Diferencia entre un dense index y un sparse index.**
8. **¿Por qué los secondary indexes son siempre densos?**
9. **¿Qué es un multilevel index, y por qué se utiliza?**
10. **Describe los beneficios y desventajas de usar un B+ tree index.**

Respuestas:

1. La data en un hard disk se organiza en círculos concéntricos llamados **tracks**. Cada track se divide en **sectors**, que son las unidades más pequeñas de data que se

pueden leer o escribir. **Blocks**, que consisten en una secuencia contigua de sectors, se utilizan para transferir data entre el disco y la memoria principal.

2. El almacenamiento de registros de longitud fija tiene como objetivo almacenar el registro i comenzando en el byte n (i -1), donde n es el tamaño fijo del registro. Sin embargo, esto puede llevar a que los registros abarquen múltiples blocks. Para evitar esto, el enfoque se modifica para asegurar que cada registro quepa dentro de un solo block.
3. Dos enfoques para manejar la eliminación de registros son: **(1) Desplazamiento de registros:** Mover los registros subsiguientes para llenar el espacio creado por el registro eliminado. **(2) Listas enlazadas:** En lugar de mover registros, enlazar espacios libres usando punteros, comenzando con un encabezado que apunta al primer registro libre y cada registro libre apuntando al siguiente.
4. **La organización de archivos heap** permite que los registros se coloquen en cualquier lugar del archivo donde haya espacio, mientras que **la organización de archivos secuencial** almacena los registros en un orden específico basado en una clave de búsqueda.
5. La organización de archivos secuencial requiere recorrer todo el archivo en el orden especificado. Al realizar un natural join, esto significa potencialmente leer ambas tablas múltiples veces para encontrar registros coincidentes, lo que lleva a ineficiencia.
6. Un **index** en una base de datos es una estructura de datos que mejora la velocidad de recuperación de datos. La compensación es que los índices requieren espacio de almacenamiento adicional y deben actualizarse cada vez que se modifica la data que indexan.
7. Un **dense index** contiene una entrada para cada valor de clave de búsqueda en el archivo de datos, mientras que un **sparse index** contiene entradas solo para un subconjunto de los valores de clave de búsqueda.
8. Los secondary indexes son siempre densos porque no determinan el orden físico del archivo de datos. Por lo tanto, para localizar cualquier registro basado en la clave secundaria, se requiere una entrada de índice para cada valor posible.
9. Un **multilevel index** es esencialmente un índice a otro índice. Se utiliza cuando el índice primario es demasiado grande para caber en la memoria, mejorando el rendimiento de búsqueda al reducir el número de accesos al disco.
10. **Los B+ tree indexes** ofrecen operaciones rápidas de búsqueda, inserción y eliminación debido a su estructura de árbol balanceada. Eliminan la necesidad de reorganización periódica. Sin embargo, pueden ser menos eficientes que otras estructuras para ciertos tipos de consultas, como consultas de rango sobre atributos no indexados.

Glosario de Términos Clave:

- **Track:** Un camino circular en un disco donde se graba la data magnéticamente.
- **Sector:** Una subdivisión de un track, que representa la unidad más pequeña de data que se puede leer o escribir desde un disco.

- **Block:** Una secuencia contigua de sectors, formando la unidad básica de transferencia de data entre disco y memoria.
- **Heap file organization:** Un método de organización de archivos donde los registros se almacenan sin un orden particular.
- **Sequential file organization:** Un método de organización de archivos donde los registros se almacenan en secuencia basada en una clave de búsqueda.
- **Index:** Una estructura de datos utilizada para acelerar la recuperación de datos proporcionando un mecanismo de búsqueda rápida para encontrar registros basados en atributos específicos.
- **Dense index:** Un índice que contiene una entrada para cada valor de clave de búsqueda en el archivo de datos.
- **Sparse index:** Un índice que contiene entradas solo para un subconjunto de valores de clave de búsqueda, típicamente utilizado cuando el archivo de datos está ordenado secuencialmente en el atributo indexado.
- **Multilevel index:** Un índice a otro índice, utilizado para mejorar el rendimiento de búsqueda para índices grandes al reducir accesos al disco.
- **B+ tree index:** Una estructura de datos de árbol balanceado comúnmente utilizada para indexación en bases de datos, ofreciendo operaciones eficientes de búsqueda, inserción y eliminación.
- **Prefix Compression:** Una técnica utilizada para acortar claves de índice al almacenar solo un prefijo de la clave, reduciendo los requisitos de espacio y mejorando la eficiencia.

2 Guía de Estudio de Álgebra Relacional

2.1 Preguntas de Respuesta Corta

Instrucciones: Responde las siguientes preguntas en 2-3 oraciones cada una.

1. **¿Qué es un operador de álgebra relacional?**
2. **¿Cómo se puede traducir una consulta SQL en una expresión en álgebra relacional?**
3. **¿Cuáles son los dos tipos de operadores de álgebra relacional? Da un ejemplo de cada uno.**
4. **Explica el concepto de "costo" en el contexto de las operaciones de álgebra relacional. ¿Cómo se mide típicamente?**
5. **¿Qué representa la notación "R = (A1::T1, ..., An::Tn)" en el contexto de esquemas relacionales?**
6. **Define la operación de proyección generalizada en álgebra relacional. Proporciona un ejemplo.**
7. **¿Cuál es el propósito de la operación de selección en álgebra relacional? ¿Cómo se denota?**

8. **Describe el algoritmo de búsqueda lineal para la operación de selección y establece su costo estimado.**
9. **¿Cuál es el propósito de un factor de selectividad en la estimación del tamaño de los resultados de las operaciones de álgebra relacional?**
10. **Explica el concepto de "join natural" en álgebra relacional. Proporciona un ejemplo.**

2.2 Clave de Respuestas Cortas

1. Un operador de álgebra relacional es una función que toma una o más relaciones (tablas) como entrada y produce una nueva relación como salida. Estos operadores permiten la manipulación y recuperación de datos dentro de una base de datos relacional.
2. Una consulta SQL se puede traducir en una expresión en álgebra relacional descomponiendo la consulta en componentes más pequeños, cada uno representando una operación específica de álgebra relacional. Estas operaciones se combinan para formar una expresión que representa la consulta original.
3. Los dos tipos de operadores de álgebra relacional son operadores lógicos y operadores físicos. Un operador lógico define la operación a realizar (por ejemplo, selección, proyección). Un operador físico define cómo se implementa la operación lógica (por ejemplo, búsqueda lineal, escaneo de índice).
4. "Costo" en álgebra relacional se refiere a los recursos utilizados al realizar una operación. Se mide típicamente en términos del número de transferencias de bloques de disco, ya que el I/O de disco es a menudo el aspecto más que consume tiempo. Esta medición ayuda a comparar la eficiencia de diferentes algoritmos.
5. Esta notación representa un esquema relacional para una relación llamada "R". Consiste en atributos A1 a An, cada uno con su correspondiente tipo de dato T1 a Tn. Por ejemplo, "Nombre::string" indica un atributo "Nombre" con el tipo de dato "string".
6. La proyección generalizada extiende la proyección estándar al permitir cálculos sobre atributos. Se denota como Π f1,...,fn(R), donde f1 a fn son funciones aplicadas a cada tupla de la relación R. Por ejemplo, para obtener los salarios anuales de los profesores de una tabla "Profesor(Nombre, Salario)", podemos usar Π Nombre, Salario*12(Profesor).
7. La operación de selección recupera tuplas de una relación que satisfacen una condición dada (predicado). Se denota como σ P(R), donde P es el predicado aplicado a cada tupla de la relación R. Por ejemplo, para seleccionar profesores con un salario mayor a 60000, usaríamos σ Salario $>$ 60000(Profesor).
8. El algoritmo de búsqueda lineal escanea cada bloque de la relación y verifica si cada tupla satisface la condición de selección. Su costo estimado es br transferencias de bloques + 1 acceso a bloque, donde br es el número de bloques que contienen tuplas en la relación R.

9. Un factor de selectividad estima la fracción de tuplas en una relación que satisfacen una condición dada. Se utiliza para estimar el tamaño de los resultados intermedios y finales después de operaciones como selección o join. Esto ayuda a optimizar los planes de ejecución de consultas.
10. Un join natural combina tuplas de dos relaciones basándose en la igualdad de todos los atributos con el mismo nombre en ambas relaciones. Por ejemplo, si tenemos relaciones R(A, B, C) y S(B, C, D), el join natural $R \bowtie S$ devolverá tuplas donde R.B = S.B y R.C = S.C.

2.3 Glosario de Términos Clave

2.3.1 Relación

Una tabla con filas (tuplas) y columnas (atributos), que representa un conjunto de datos relacionados.

2.3.2 Esquema

Define la estructura de una relación, especificando el nombre de cada atributo y su tipo de dato.

2.3.3 Tupla

Una fila en una relación, que representa una única instancia de los datos descritos por el esquema.

2.3.4 Atributo

Una columna nombrada en una relación, que representa una característica específica de los datos.

2.3.5 Dominio

El conjunto de valores posibles que un atributo puede tomar.

2.3.6 Álgebra Relacional

Un lenguaje de consulta formal que proporciona un conjunto de operadores para manipular relaciones.

2.3.7 Operador Lógico

Un operador de álgebra relacional que define la operación a realizar, como selección, proyección o join.

2.3.8 Operador Físico

Una implementación concreta de un operador lógico, especificando los algoritmos y estructuras de datos utilizados para realizar la operación.

2.3.9 Optimización de Consultas

El proceso de encontrar la forma más eficiente de ejecutar una consulta dada, típicamente considerando diferentes planes de ejecución y eligiendo el que tenga el costo estimado más bajo.

2.3.10 Transferencia de Bloque de Disco

El movimiento de un bloque de datos entre disco y memoria. El número de transferencias de bloques se utiliza a menudo como medida del costo de una operación de álgebra relacional.

2.3.11 Índice B+ Tree

Una estructura de datos en forma de árbol que permite la búsqueda, inserción y eliminación eficientes de datos. Los índices B+ tree se pueden utilizar para optimizar operaciones de álgebra relacional proporcionando acceso rápido a tuplas basadas en los valores de los atributos indexados.

2.3.12 Factor de Selectividad

Una medida de la fracción de tuplas en una relación que satisfacen una condición dada. Se utiliza para estimar el tamaño de los resultados de las operaciones de álgebra relacional.

2.3.13 Join de Bucle Anidado

Un algoritmo de join que itera sobre cada tupla en la relación externa y para cada tupla, escanea toda la relación interna para encontrar tuplas coincidentes.

2.3.14 Merge-Join

Un algoritmo de join que ordena ambas relaciones en los atributos de join y luego escanea las relaciones ordenadas una vez para encontrar tuplas coincidentes.

2.3.15 Eliminación de Duplicados

El proceso de eliminar tuplas duplicadas de una relación.

2.3.16 Ordenamiento Externo

Un algoritmo de ordenamiento diseñado para manejar conjuntos de datos que son demasiado grandes para caber en memoria. Los algoritmos de ordenamiento externo típicamente implican dividir los datos en fragmentos más pequeños, ordenar los fragmentos en memoria y luego fusionar los fragmentos ordenados para producir el resultado final ordenado.

2.3.17 Concatenación (Union All)

Una operación de álgebra relacional que combina las tuplas de dos relaciones sin eliminar duplicados.

2.3.18 Intersección

Una operación de álgebra relacional que devuelve solo las tuplas que están presentes en ambas relaciones de entrada.

2.3.19 Diferencia (Minus)

Una operación de álgebra relacional que devuelve las tuplas que están presentes en la primera relación pero no en la segunda relación.

3 Guía de Estudio de Álgebra Relacional y Optimización de Consultas

3.1 Preguntas de Respuesta Corta

1. **¿Qué es un query evaluation plan y cómo se relaciona con la optimización de consultas?**
2. **Explica la diferencia entre materialización y pipelining en la evaluación de consultas.**
3. **¿Cuál es el propósito del selectivity factor en la optimización de consultas y cómo se calcula para una operación de selección?**
4. **Describe el concepto de "pushing selections" en la optimización de consultas y explica sus beneficios.**
5. **Explica por qué el orden de los natural joins en una consulta puede impactar significativamente el rendimiento.**
6. **¿Qué es un left-deep join tree y cómo se diferencia de otros árboles de unión?**
7. **¿Por qué se utilizan a menudo heurísticas en la optimización de consultas en lugar de siempre buscar el plan absoluto óptimo?**
8. **Da un ejemplo de una regla heurística utilizada para la selección del orden de unión y explica su razonamiento.**
9. **¿Cuáles son las ventajas de usar programación dinámica para la optimización del orden de unión en comparación con un enfoque de fuerza bruta?**
10. **Describe brevemente cómo los enfoques híbridos combinan heurísticas y optimización basada en costos en la optimización de consultas.**

3.2 Clave de Respuestas Cortas

1. Un query evaluation plan describe los pasos específicos y algoritmos utilizados para ejecutar una consulta. La optimización de consultas tiene como objetivo encontrar el plan más eficiente considerando diferentes equivalencias lógicas y operadores físicos.
2. La materialización almacena resultados intermedios en disco como tablas temporales, mientras que el pipelining pasa directamente los resultados entre operadores sin almacenarlos. La materialización utiliza menos memoria pero más espacio en disco, mientras que el pipelining conserva espacio en disco pero requiere más memoria.
3. El selectivity factor estima la fracción de tuplas que satisfacen una condición de selección. Para la selección σ (P, r), se calcula como el número de tuplas que satisfacen P dividido por el número total de tuplas en r.

4. “Pushing selections” aplica operaciones de selección tan pronto como sea posible en el árbol de consultas. Esto reduce el tamaño de las relaciones intermedias, mejorando el rendimiento general de la consulta.
5. Diferentes órdenes de unión producen relaciones intermedias de tamaños variables. Optimizar el orden de unión minimiza el tamaño de los resultados intermedios, lo que lleva a una ejecución de consulta más rápida.
6. Un left-deep join tree tiene la propiedad de que el operando derecho de cada unión es siempre una relación base, no el resultado de una unión intermedia. Esta estructura se prefiere a menudo para la evaluación en pipelining.
7. Encontrar el plan de consulta absoluto óptimo puede ser computacionalmente costoso, especialmente para consultas complejas. Las heurísticas ofrecen buenas soluciones rápidamente, incluso si no son necesariamente las mejores.
8. Una heurística es unir las relaciones más pequeñas primero. Esto minimiza el tamaño inicial de la relación intermedia, lo que puede llevar a uniones subsiguientes más rápidas.
9. La programación dinámica descompone el problema de encontrar el orden óptimo de unión en subproblemas más pequeños y almacena sus soluciones. Esto evita cálculos redundantes y reduce significativamente el espacio de búsqueda en comparación con un enfoque de fuerza bruta.
10. Los enfoques híbridos aplican heurísticas para partes de la consulta y utilizan optimización basada en costos para otras, equilibrando el rendimiento y el costo de optimización. Por ejemplo, pueden usar heurísticas para el orden de unión dentro de subconsultas específicas mientras emplean un enfoque basado en costos para optimizar entre subconsultas.

3.3 Glosario

3.3.1 Query Evaluation Plan

Una secuencia de pasos y algoritmos utilizados por un sistema de gestión de bases de datos para ejecutar una consulta. Describe el orden de operaciones, métodos de acceso y algoritmos para cada operador.

3.3.2 Query Optimization

El proceso de encontrar la forma más eficiente de ejecutar una consulta de base de datos, considerando factores como el tiempo de ejecución, la utilización de recursos y el costo general.

3.3.3 Materialization

Una estrategia de evaluación donde los resultados intermedios de una consulta se almacenan en disco como tablas temporales. Este enfoque reduce el uso de memoria pero puede aumentar el I/O de disco.

3.3.4 Pipelining

Una estrategia de evaluación donde los resultados de un operador se pasan directamente como entrada al siguiente operador sin almacenarlos como relaciones temporales. Este enfoque minimiza el I/O de disco pero requiere más memoria para almacenar en búfer los resultados intermedios.

3.3.5 Selectivity Factor

Una medida estadística utilizada en la optimización de consultas para estimar la fracción de tuplas en una relación que satisfarán un predicado de selección dado.

3.3.6 "Pushing Selections"

Una técnica de optimización de consultas que aplica selecciones tan pronto como sea posible en el árbol de consultas para reducir el tamaño de las relaciones intermedias y mejorar el rendimiento.

3.3.7 Join Order

El orden en que se unen las relaciones en una consulta que involucra múltiples tablas. Elegir un orden de unión eficiente puede impactar significativamente el rendimiento de la consulta.

3.3.8 Left-Deep Join Tree

Un árbol de consulta que representa uniones donde el operando derecho de cada unión es siempre una relación base, no un resultado intermedio. Esta estructura se prefiere a menudo para la evaluación en pipelining.

3.3.9 Heuristic

Una regla general o directriz utilizada en la optimización de consultas para tomar decisiones rápidas y encontrar planes de ejecución buenos, pero no necesariamente óptimos. Las heurísticas se utilizan a menudo para reducir el espacio de búsqueda para la optimización.

3.3.10 Cost-Based Optimization

Un enfoque de optimización de consultas que asigna costos a diferentes planes de ejecución basados en factores como I/O de disco, tiempo de CPU y costos de comunicación. El optimizador elige el plan con el costo estimado más bajo.

3.3.11 Dynamic Programming

Una técnica utilizada en la optimización de consultas para encontrar el orden óptimo de unión para una consulta descomponiendo el problema en subproblemas más pequeños y superpuestos y almacenando sus soluciones para evitar cálculos redundantes.

3.3.12 Hybrid Query Optimization

Una combinación de técnicas de optimización basadas en heurísticas y en costos. Los enfoques híbridos buscan equilibrar la eficiencia de la optimización con la calidad de los planes de ejecución generados.

4 Guía de Estudio de Recuperación de Información

4.1 Preguntas de Respuesta Corta

Instrucciones: Responde las siguientes preguntas en 2-3 oraciones cada una.

1. ¿Qué es information retrieval (IR) y qué tipo de datos maneja típicamente?
2. Explica tres diferencias clave entre bases de datos relacionales y sistemas de IR.
3. Describe dos tipos diferentes de lenguajes de consulta utilizados en sistemas de IR.
4. ¿Cuál es el propósito de usar proximity operators en consultas de IR? Proporciona un ejemplo.
5. ¿Cómo maneja Google Search las consultas en lenguaje natural y cuáles son algunos operadores que utiliza?
6. ¿Cuáles son los principales factores que determinan la relevancia de un documento para una consulta?

7. Describe brevemente el modelo Boolean de recuperación de información. ¿Cuáles son sus limitaciones?
8. ¿Cómo representa el modelo de espacio vectorial documentos y consultas? ¿Cómo se mide la relevancia en este modelo?
9. ¿Qué es TF-IDF y cómo se utiliza en el modelo de espacio vectorial? Explica la idea detrás de esto.
10. Explica el propósito y el proceso de stemming y synonym handling en el contexto de IR.

4.2 Clave de Respuestas Cortas

1. Information retrieval (IR) es el proceso de encontrar documentos de una colección que son relevantes para la consulta de un usuario. Los sistemas de IR manejan típicamente datos no estructurados en lenguaje natural.
2. Tres diferencias clave: a) Las bases de datos relacionales manejan datos estructurados en tablas, mientras que los sistemas de IR manejan datos no estructurados como texto. b) Las bases de datos relacionales utilizan lenguajes de consulta estructurados (SQL), mientras que los sistemas de IR a menudo emplean consultas basadas en palabras clave o en lenguaje natural. c) Las bases de datos relacionales devuelven coincidencias exactas, mientras que los sistemas de IR clasifican los resultados según la relevancia.
3. Dos tipos de lenguajes de consulta: a) Consultas Booleanas, donde los usuarios combinan palabras clave con operadores como AND, OR y NOT. b) Consultas en lenguaje natural, donde los usuarios expresan su necesidad de información en lenguaje cotidiano, como hacer una pregunta.
4. Los proximity operators especifican cuán cerca deben estar ciertos términos entre sí en un documento. Por ejemplo, "ciencia NEAR tecnología" recuperaría documentos donde estos términos aparecen juntos, sugiriendo una relación más fuerte entre los conceptos.
5. Google Search utiliza un algoritmo sofisticado para entender la intención y el significado detrás de las consultas en lenguaje natural. También utiliza operadores como comillas para coincidencias exactas, OR para términos alternativos, "-" para exclusión, "site:" para limitar resultados a un sitio web específico, y "intitle:" para buscar dentro de los títulos de las páginas.
6. La relevancia se determina por factores como: a) Frecuencia de término: cuán a menudo aparecen los términos de consulta en un documento. b) Frecuencia inversa de documento: cuán raro es un término en toda la colección de documentos (los términos más raros se consideran más informativos). c) Enlaces a un documento: el número y la calidad de los enlaces que apuntan a una página pueden indicar su importancia y autoridad.
7. El modelo Boolean representa documentos como conjuntos de términos y utiliza operadores Boolean (AND, OR, NOT) para construir consultas. Los resultados son una coincidencia o no, sin clasificación. Las limitaciones incluyen la dificultad para

expresar necesidades de información complejas y la falta de noción de coincidencias parciales o clasificación de relevancia.

8. El modelo de espacio vectorial representa tanto documentos como consultas como vectores en un espacio multidimensional, donde cada dimensión corresponde a un término. La relevancia se mide mediante la similitud del coseno entre el vector de consulta y los vectores de documentos, con una mayor similitud que indica una mayor relevancia.
9. TF-IDF significa Term Frequency-Inverse Document Frequency. Es un esquema de ponderación utilizado en el modelo de espacio vectorial para evaluar la importancia de un término dentro de un documento y en toda la colección. La idea es que los términos que aparecen con frecuencia en un documento pero raramente en otros son más informativos y discriminativos.
10. Stemming reduce las palabras a su forma raíz (por ejemplo, "corriendo", "corre" se convierten en "correr"), agrupando palabras similares. El synonym handling implica el uso de un tesauro u ontología para considerar términos alternativos que transmiten el mismo significado, mejorando la precisión y el recuerdo de la recuperación.

4.3 Glosario de Términos Clave

Modelo Boolean: Un modelo de IR que representa documentos y consultas como conjuntos de términos, utilizando operadores Boolean (AND, OR, NOT) para la recuperación.

Similitud del Coseno: Una medida de similitud entre dos vectores, a menudo utilizada en el modelo de espacio vectorial para determinar la relevancia de un documento para una consulta.

Recuperación de Información (IR): El proceso de encontrar información relevante de una colección de documentos en respuesta a la consulta de un usuario.

Inverted Index: Una estructura de datos utilizada en sistemas de IR que mapea términos a los documentos que los contienen, permitiendo una recuperación eficiente.

Consulta en Lenguaje Natural: Una consulta expresada en lenguaje cotidiano, permitiendo a los usuarios interactuar con sistemas de IR de manera más intuitiva.

Precisión: Una métrica en IR que mide la proporción de documentos recuperados que son relevantes para la consulta.

Proximity Operator: Un operador de búsqueda utilizado para especificar cuán cerca deben estar ciertos términos entre sí en un documento recuperado (por ejemplo, NEAR, WITHIN).

Recall: Una métrica en IR que mide la proporción de documentos relevantes que son recuperados con éxito.

Relevancia: El grado en que un documento satisface la necesidad de información expresada en la consulta de un usuario.

Stemming: El proceso de reducir palabras a su forma raíz para mejorar la eficiencia y efectividad de la recuperación.

Stop Words: Palabras comunes (por ejemplo, "el", "una", "es") que a menudo son eliminadas de documentos y consultas ya que tienen poco valor informativo.

Synonym Handling: El uso de un tesauro u ontología para considerar términos alternativos con significados similares durante la recuperación, mejorando el recall.

TF-IDF (Term Frequency-Inverse Document Frequency): Un esquema de ponderación utilizado en IR para evaluar la importancia de un término dentro de un documento y en toda la colección.

Modelo de Espacio Vectorial: Un modelo de IR que representa documentos y consultas como vectores en un espacio multidimensional, utilizando la similitud del coseno para medir la relevancia.

5 Guía de Estudio de Recuperación de Información Web

5.0.1 Glosario de Términos Clave

Web Crawler Un programa que navega sistemáticamente por la World Wide Web, típicamente con el propósito de indexación web.

Indexing El proceso de crear una estructura de datos que permite una búsqueda eficiente de una colección de documentos.

Inverted Index Una estructura de datos que mapea palabras a los documentos en los que aparecen, facilitando búsquedas rápidas por palabras clave.

TF-IDF Frecuencia de Término-Frecuencia Inversa de Documento: una estadística numérica que refleja cuán importante es una palabra para un documento en una colección. Se utiliza para la recuperación de información y la minería de texto.

PageRank Un algoritmo de Google que asigna un peso numérico a cada elemento de un conjunto de documentos hiperconectados, como la World Wide Web, con el propósito de "medir" su importancia relativa dentro del conjunto.

Anchor Text El texto visible y clicable en un hipervínculo.

Hits Algorithm Un algoritmo que mide la importancia de una página web en función del número de otras páginas importantes que enlazan a ella.

Snippet Un breve extracto de texto de una página web, mostrado en una página de resultados de motores de búsqueda (SERP) para dar al usuario una vista previa del contenido de la página.

Breadcrumb Una ayuda de navegación que muestra al usuario el camino que ha tomado para llegar a su ubicación actual en un sitio web.

Knowledge Panel Una caja que aparece en el lado derecho de la página de resultados de búsqueda de Google, proporcionando un resumen de información sobre un tema, persona, lugar o cosa en particular.

5.0.2 Preguntas

Instrucciones: Responde las siguientes preguntas en 2-3 oraciones cada una.

1. ¿Cuál es la diferencia fundamental entre un motor de búsqueda web y un motor de búsqueda tradicional?
2. Explica cómo los web crawlers utilizan hipervínculos para descubrir nuevas páginas web.
3. ¿Por qué es crucial que los motores de búsqueda web tengan sistemas de indexación robustos?
4. Describe los desafíos que plantean las páginas de spam a los motores de búsqueda web y cómo se pueden abordar.
5. ¿Por qué resultó insuficiente confiar únicamente en TF-IDF para clasificar las páginas web de manera efectiva?
6. Explica el concepto de "site popularity" y cómo contribuye al ranking de resultados de búsqueda.
7. ¿Cómo utiliza PageRank un modelo de caminata aleatoria para calcular la importancia de las páginas web?
8. ¿Cuál es la importancia del anchor text en la evaluación de la relevancia de las páginas web para temas específicos?
9. ¿Cómo utiliza Google el comportamiento de búsqueda de los usuarios para mejorar la precisión y relevancia de sus resultados de búsqueda?
10. Describe los diferentes tipos de información presentados en un resultado típico de búsqueda de Google, más allá de solo un enlace a la página web.

5.0.3 Respuestas Cortas

1. Los motores de búsqueda tradicionales operan sobre una colección de documentos predefinida, mientras que los motores de búsqueda web descubren e indexan constantemente nuevas páginas en la siempre cambiante World Wide Web.
2. Los web crawlers comienzan con un conjunto de páginas semilla y siguen los hipervínculos presentes en esas páginas para encontrar nuevos documentos. Este proceso continúa recursivamente, expandiendo la colección de páginas indexadas.

3. Los sistemas de indexación organizan y categorizan las páginas web en función de su contenido, permitiendo una recuperación rápida y eficiente de páginas relevantes en respuesta a consultas de usuarios.
4. Las páginas de spam manipulan los algoritmos de los motores de búsqueda al llenar de palabras clave o crear enlaces artificiales. Los motores de búsqueda combaten esto a través de algoritmos que identifican y penalizan prácticas de spam, y mediante la incorporación de retroalimentación de usuarios.
5. La vastedad de la web y la existencia de páginas de spam significaban que TF-IDF por sí solo no podía reflejar con precisión la verdadera relevancia o autoridad de una página web.
6. La popularidad del sitio se refiere a métricas como el número de visitas que recibe un sitio web o cuántos otros sitios web enlazan a él. Sirve como un proxy para la calidad y confiabilidad de un sitio, influyendo en su clasificación en los resultados de búsqueda.
7. PageRank simula un surfista web aleatorio que hace clic en enlaces al azar. La probabilidad de aterrizar en una página particular refleja su PageRank, siendo las páginas que reciben más enlaces de páginas de alto rango las que tienen una puntuación más alta.
8. El anchor text proporciona pistas contextuales sobre el contenido de la página enlazada. Los motores de búsqueda utilizan esta información para evaluar la relevancia de la página para consultas que contienen esas palabras clave.
9. Google rastrea clics de usuarios, tiempo de permanencia y otras interacciones con los resultados de búsqueda. Estos datos ayudan a refinar los algoritmos de clasificación, promoviendo páginas que los usuarios encuentran genuinamente útiles y desalentando aquellas que no logran involucrar a los usuarios.
10. Un resultado de búsqueda de Google típicamente incluye el título de la página, URL, un snippet de texto relevante, breadcrumbs, y a veces información adicional como la fecha, sitelinks, o rich snippets (por ejemplo, imágenes, reseñas).

6 Guía de Estudio de Procesamiento de Lenguaje Natural

6.1 Glosario de Términos Clave

6.1.1 Procesamiento de Lenguaje Natural (NLP)

Un subcampo de la inteligencia artificial (AI) que se centra en permitir que las computadoras comprendan, interpreten y generen lenguaje humano.

6.1.2 Análisis Sintáctico

El estudio de la estructura gramatical de las oraciones, identificando partes del discurso (sustantivos, verbos, etc.) y sus relaciones.

6.1.3 Análisis Semántico

Se centra en el significado de las palabras y frases en contexto, permitiendo que las máquinas comprendan el contenido.

6.1.4 Reconocimiento de Entidades Nombradas (NER)

Identifica y clasifica entidades dentro del texto, como nombres de personas, lugares u organizaciones.

6.1.5 Generación de Lenguaje Natural (NLG)

Permite a las máquinas generar texto coherente y relevante a partir de datos estructurados.

6.1.6 Análisis de Sentimientos

Evalúa el tono emocional del texto, determinando si es positivo, negativo o neutral.

6.1.7 Comprensión del Lenguaje Natural (NLU)

Un subcampo de AI centrado en entrenar a las máquinas para comprender e interpretar el lenguaje humano de manera significativa, incluyendo contexto, intención y sentimiento.

6.1.8 Tokenización

Descomponer el texto en unidades más pequeñas, como palabras o frases.

6.1.9 Stop Words

Palabras comunes que no contribuyen con un significado significativo al análisis, como “y”, “el”, “de”.

6.1.10 Stemming

Reducir las palabras a su forma raíz eliminando sufijos y prefijos.

6.1.11 Lematización

Similar al stemming, pero devuelve la forma base o raíz de una palabra considerando su contexto gramatical.

6.1.12 Etiquetado de Partes del Discurso

Asignar etiquetas gramaticales a cada token, como sustantivo, verbo, adjetivo, etc.

6.1.13 Contexto

La información circundante que ayuda a interpretar el significado de una palabra, frase o conversación en NLP.

6.1.14 Contexto Local

El entorno inmediato de una palabra o frase dentro de una oración.

6.1.15 Contexto Global

Información que se extiende más allá de una sola oración o párrafo, incluyendo el historial de interacciones.

6.1.16 Contexto Conversacional

La dinámica del diálogo entre los participantes, incluyendo turnos de habla y respuestas anteriores.

6.1.17 Contexto Situacional

Factores externos que influyen en la interpretación del lenguaje, como el entorno físico o el estado emocional.

6.1.18 Contexto Histórico

Conocimiento acumulado sobre un tema o preferencias del usuario a lo largo del tiempo.

6.1.19 Contexto Cultural

Factores sociales y culturales que influyen en la interpretación del lenguaje, afectando tanto la forma como el contenido.

6.1.20 Resolución de Correferencias

Identificar a qué entidades se refieren los pronombres o expresiones en un texto.

6.1.21 Análisis Pragmático

Considerar el contexto situacional, la intención del hablante y las implicaciones sociales para interpretar el lenguaje en situaciones específicas.

6.2 Cuestionario Corto

6.2.1 ¿Qué es el Procesamiento de Lenguaje Natural (NLP) y cuál es su objetivo principal?

El Procesamiento de Lenguaje Natural (NLP) es un subcampo de la inteligencia artificial que se centra en permitir que las computadoras comprendan, interpreten y generen lenguaje humano. Su objetivo principal es cerrar la brecha de comunicación entre humanos y máquinas, permitiendo interacciones más naturales y efectivas.

6.2.2 Diferencie entre Análisis Sintáctico y Análisis Semántico en NLP. Proporcione un ejemplo para cada uno.

El análisis sintáctico examina la estructura gramatical de las oraciones, identificando partes del discurso y sus relaciones (por ejemplo, "El gato se sentó en la alfombra" - sujeto (gato), verbo (sentó), preposición (en), objeto (alfombra)). **El análisis semántico** se centra en comprender el significado de las palabras y frases en contexto (por ejemplo, "El banco aprobó mi préstamo" - 'banco' se refiere a una institución financiera, no a un banco de río, según el contexto).

6.2.3 Explique la función del Reconocimiento de Entidades Nombradas (NER) con un ejemplo.

El Reconocimiento de Entidades Nombradas (NER) identifica y clasifica entidades dentro del texto, como personas, lugares, organizaciones y fechas. Por ejemplo, en la oración "Apple Inc. tiene su sede en Cupertino, California," NER identificaría "Apple Inc." como una organización y "Cupertino, California" como una ubicación.

6.2.4 ¿Cuáles son las principales diferencias entre Comprensión del Lenguaje Natural (NLU) y Generación de Lenguaje Natural (NLG)?

NLU se centra en permitir que las computadoras comprendan e interpreten el lenguaje humano, incluyendo contexto, intención y sentimiento, mientras que **NLG** se centra en permitir que las computadoras generen texto similar al humano a partir de datos estructurados. NLU se trata de comprensión, mientras que NLG se trata de creación.

6.2.5 Describa dos aplicaciones comunes de NLU en la vida cotidiana.

Dos aplicaciones comunes de **NLU** son los asistentes de voz (como Siri o Alexa), que utilizan NLU para comprender y responder a comandos de voz, y los chatbots, que utilizan NLU para interactuar con los usuarios en tiempo real, proporcionando soporte al cliente o información.

6.2.6 Explique brevemente el propósito e importancia del pre-procesamiento en NLP.

El pre-procesamiento prepara los datos de texto para el análisis limpiándolos, estandarizándolos y transformándolos en un formato que los algoritmos puedan entender. Esto es importante porque los datos de texto en bruto suelen ser desordenados e inconsistentes, lo que puede obstaculizar un análisis preciso de NLP.

6.2.7 ¿Qué son las "stop words" en NLP y por qué se eliminan típicamente durante el pre-procesamiento?

Las "stop words" son palabras comunes como "y", "el" y "de" que se eliminan típicamente durante el pre-procesamiento porque no contribuyen con un significado significativo al análisis. Eliminar las stop words reduce el volumen de datos y mejora la eficiencia del procesamiento al centrarse en palabras significativas.

6.2.8 ¿Cuál es la diferencia entre Stemming y Lematización? Proporcione un ejemplo para cada uno.

El Stemming reduce las palabras a su forma raíz eliminando sufijos y prefijos (por ejemplo, "corriendo" se convierte en "correr"). **La Lematización**, sin embargo, considera el contexto gramatical para devolver la forma base o raíz de una palabra (por ejemplo, "mejor" se convierte en "bueno"). El stemming es más rápido pero puede producir no-palabras, mientras que la lematización es más lenta pero más precisa.

6.2.9 Defina "contexto" en NLP y explique por qué es crucial para una interpretación precisa del lenguaje.

En NLP, **"contexto"** se refiere a la información circundante que ayuda a interpretar el significado de una palabra, frase o conversación. Es crucial para una interpretación precisa del lenguaje porque las palabras pueden tener múltiples significados y su significado pretendido a menudo se determina por el contexto en el que se utilizan.

6.2.10 Describa dos tipos de contexto en NLP y proporcione un ejemplo de cómo cada tipo influye en la comprensión del lenguaje.

Dos tipos de **contexto** son **contexto local**, que es el entorno inmediato de una palabra dentro de una oración (por ejemplo, en "El banco está cerrado," el contexto local ayuda a determinar si "banco" se refiere a una institución financiera o a un banco de río) y **contexto global**, que abarca información que se extiende más allá de una sola oración, como el historial de interacciones (por ejemplo, si un usuario mencionó previamente un producto específico, el sistema puede usar esa información para comprender preguntas posteriores sobre él).

7 Guía de Estudio de Ingeniería de Prompts

7.1 Cuestionario de Respuesta Corta

7.1.1 1. ¿Qué es un prompt en el contexto del Procesamiento de Lenguaje Natural (NLP)?

Un prompt es una instrucción o pregunta proporcionada a un modelo de NLP para guiar su respuesta y especificar el formato de salida deseado.

7.1.2 2. Explique el concepto de ingeniería de prompts y su importancia en NLP.

La ingeniería de prompts implica el diseño y optimización hábil de prompts para obtener resultados precisos y eficientes de los modelos de NLP. Incluye refinar prompts, probar enfoques y ajustar parámetros del modelo para un rendimiento óptimo.

7.1.3 3. Describa tres modificaciones específicas que caen bajo el paraguas de la ingeniería de prompts.

Las modificaciones en la ingeniería de prompts pueden incluir ajustar parámetros del modelo, incorporar datos de entrenamiento relevantes, refinar la estructura del prompt para una mejor comprensión y experimentar con diferentes formulaciones de prompts.

7.1.4 4. ¿Qué es la tokenización y por qué es relevante para trabajar con modelos de lenguaje grandes?

La tokenización es el proceso de descomponer la entrada de texto en unidades individuales llamadas tokens, que son secuencias de caracteres tratadas como una sola entidad. La tokenización es importante porque los costos de uso de modelos de lenguaje a menudo se basan en el número de tokens procesados.

7.1.5 5. ¿Cómo afecta el parámetro "temperatura" la salida de un modelo de lenguaje en el OpenAI playground?

El parámetro "temperatura" controla la aleatoriedad de la salida de un modelo de lenguaje. Una temperatura más alta conduce a respuestas más diversas y creativas, pero potencialmente menos precisas. Una temperatura más baja produce salidas más predecibles y deterministas, pero potencialmente más confiables.

7.1.6 6. Explique cómo el parámetro "TopP" influye en el rango creativo y la precisión de las respuestas de un modelo de lenguaje.

"TopP" (muestreo de núcleo) restringe el rango de opciones de texto consideradas por el modelo. Un valor alto de TopP permite salidas más diversas pero potencialmente menos relevantes, mientras que un TopP más bajo se centra en las respuestas más probables y precisas, aunque con potencialmente menos creatividad.

7.1.7 7. Diferencie entre "Frecuencia Penalty" y "Presence Penalty" como parámetros en el OpenAI playground.

"Frecuencia Penalty" desalienta la repetición de tokens específicos (palabras o frases) en la salida del modelo, aumentando la penalización con cada repetición. "Presence Penalty", por otro lado, penaliza la aparición de cualquier token que ya se haya utilizado, independientemente de su frecuencia.

7.1.8 8. Nombra y describe los tres elementos fundamentales que componen un prompt.

Un prompt se compone de una "acción" (la tarea a realizar), "modificadores" (restricciones que dan forma a la salida) y "datos" (información específica que se debe incluir en la respuesta).

7.1.9 9. ¿Cuál es la diferencia entre un "Standard Prompt" y un "Chain of Thought" prompt?

Un "Standard Prompt" provoca una respuesta directa del modelo sin solicitar explícitamente el razonamiento detrás de ella. Un "Chain of Thought" prompt anima al modelo a generar un proceso de pensamiento paso a paso que conduzca a su respuesta, haciendo que el razonamiento sea transparente.

7.1.10 10. Explique el concepto de un "Few-Shot Prompt" y proporcione un ejemplo de su aplicación.

"Few-Shot Prompts" proporcionan al modelo un pequeño número de ejemplos etiquetados para guiar su aprendizaje y mejorar su capacidad para realizar una tarea específica. Por ejemplo, mostrar a un modelo de lenguaje unos pocos ejemplos de traducción de frases en inglés a español antes de pedirle que traduzca una nueva frase.

7.2 Glosario de Términos Clave

- **Prompt**: Una instrucción o pregunta dada a un modelo de lenguaje para guiar su salida y especificar la respuesta deseada.
- **Ingeniería de Prompts**: El proceso de diseñar y refinar prompts para optimizar el rendimiento de los modelos de lenguaje, asegurando salidas precisas, eficientes y creativas.
- **Tokenización**: El proceso de dividir el texto en unidades individuales (tokens), que a menudo son palabras o partes de palabras.
- **Temperatura**: Un parámetro que controla la aleatoriedad y creatividad de la salida de un modelo de lenguaje. Temperatura más alta = más diversa pero potencialmente menos precisa; temperatura más baja = más predecible y potencialmente más confiable.
- **TopP (Muestreo de Núcleo)**: Un parámetro que limita el rango de opciones de texto consideradas por un modelo, equilibrando creatividad y precisión.
- **Frecuencia Penalty**: Un parámetro que desalienta la repetición de tokens específicos, aumentando la penalización con cada repetición.
- **Presence Penalty**: Un parámetro que penaliza la aparición de cualquier token ya utilizado en la salida, independientemente de su frecuencia.

- **Chain of Thought (CoT)**: Una técnica de prompting que anima a un modelo de lenguaje a generar un proceso de razonamiento paso a paso que conduzca a su respuesta.
- **Zero-Shot Prompt**: Un prompt que no requiere ejemplos; el modelo intenta completar la tarea basándose en su conocimiento preexistente.
- **One-Shot Prompt**: Un prompt que proporciona un solo ejemplo para guiar el rendimiento del modelo en una tarea específica.
- **Few-Shot Prompt**: Un prompt que proporciona un pequeño número de ejemplos etiquetados para mejorar la capacidad del modelo para aprender y realizar una tarea.

8 Chatbots Conversacionales Inteligentes: Una Guía de Estudio

8.1 Cuestionario

8.1.1 1. ¿Qué son los modelos de texto y qué tareas pueden realizar? Proporcione un ejemplo de un modelo de texto bien conocido.

Respuesta: Los modelos de texto se basan en el Procesamiento de Lenguaje Natural (NLP) y están diseñados para tareas de lectura y escritura como resumen, mapeo mental y traducción. Un ejemplo prominente es el modelo GPT.

8.1.2 2. Explique el concepto de modelado de lenguaje y cómo aprovecha las relaciones predecibles entre palabras.

Respuesta: El modelado de lenguaje opera bajo el principio de que las palabras en un idioma no se utilizan al azar, sino que están relacionadas de maneras predecibles. Esto significa que en un contexto dado, ciertas palabras son más propensas a seguir a otras, lo que permite a los modelos predecir la siguiente palabra en una secuencia.

8.1.3 3. Diferencie entre Modelos de Lenguaje Grandes (LLMs) y modelos de lenguaje tradicionales.

Respuesta: Los LLMs, como GPT y BERT, se entrenan en enormes conjuntos de datos de texto y aprovechan redes neuronales profundas para procesar información y aprender patrones de lenguaje intrincados, diferenciándolos de los modelos de lenguaje tradicionales con conjuntos de datos más pequeños y arquitecturas más simples.

8.1.4 4. Describa las características clave de un modelo conversacional y proporcione un ejemplo.

Respuesta: Los modelos conversacionales son modelos de generación de texto entrenados específicamente para la conversación. Se destacan en generar respuestas que imitan la interacción humana. ChatGPT, derivado de un modelo de texto, ejemplifica un modelo conversacional o de diálogo.

8.1.5 5. Esboce las tres etapas principales involucradas en la creación de un modelo de IA.

Respuesta: Las tres etapas principales son la preparación de datos (limpieza, filtrado y organización de datos), el entrenamiento del modelo (selección de arquitectura, tokenización de datos y ejecución de procesos de entrenamiento) y la validación (evaluación del rendimiento del modelo utilizando conjuntos de datos de referencia y métricas).

8.1.6 6. ¿Cuáles son las consideraciones clave durante la fase de validación del desarrollo del modelo de IA?

Respuesta: La validación asegura que el modelo entrenado funcione de manera efectiva. Esto implica probar contra conjuntos de datos de referencia, medir la precisión de las predicciones, la similitud del texto con las referencias y la calidad de las salidas como traducciones, utilizando métricas específicas para cada tarea.

8.1.7 7. Explique el propósito y los beneficios de ajustar un modelo de IA.

Respuesta: El ajuste fino implica ajustar un modelo preentrenado para sobresalir en una tarea o dominio específico. Al entrenar en un conjunto de datos específico, se mejora el rendimiento del modelo, ofreciendo una alternativa más rápida que construir un modelo desde cero.

8.1.8 8. Describa el papel de un componente de gestión de diálogos en la arquitectura de un chatbot conversacional.

Respuesta: El componente de gestión de diálogos asegura conversaciones coherentes y atractivas. Rastrea el historial de la conversación, gestiona el estado actual del diálogo (incluyendo preferencias del usuario e interacciones previas) y aplica estrategias para mantener un flujo natural en las conversaciones.

8.1.9 9. ¿Cómo difiere un modelo de lenguaje de propósito general de un chatbot conversacional inteligente en términos de gestión de diálogos?

Respuesta: Los modelos de lenguaje de propósito general carecen de gestión de diálogos integrada y manejan cada prompt de manera independiente, sin retener el contexto de interacciones anteriores. Los chatbots conversacionales, por el contrario, incorporan estrategias de gestión de diálogos para mantener el contexto y la coherencia a lo largo de múltiples interacciones, lo que les permite proporcionar respuestas basadas en intercambios previos.

8.1.10 10. ¿Cuáles son las funciones primarias de un codificador en una arquitectura de transformador?

Respuesta: Los codificadores procesan y comprenden secuencias de texto transformando el texto de entrada en representaciones contextuales. Utilizan mecanismos como la tokenización, la codificación posicional y la autoatención para capturar el significado y las relaciones entre palabras, generando representaciones vectoriales que codifican esta información.

8.2 Glosario de Términos Clave

- **Chatbot**: Un programa de computadora que simula la conversación con usuarios humanos, especialmente a través de Internet.
- **Modelo Conversacional**: Un tipo de modelo de lenguaje entrenado específicamente para participar en conversaciones y generar respuestas similares a las humanas.
- **Redes Neuronales Profundas**: Un tipo de red neuronal artificial con múltiples capas entre las capas de entrada y salida, lo que le permite aprender patrones complejos.
- **Codificador**: Un componente en la arquitectura de transformadores que procesa secuencias de texto de entrada para generar representaciones contextuales.
- **Ajuste Fino**: El proceso de adaptar un modelo de lenguaje preentrenado a una tarea o dominio específico mediante un entrenamiento adicional en un conjunto de datos específico.
- **IA Generativa**: Un tipo de inteligencia artificial que se centra en crear nuevo contenido, como texto, imágenes o música.
- **Modelo de Lenguaje Grande (LLM)**: Un modelo de lenguaje entrenado en enormes conjuntos de datos de texto, capaz de generar texto de calidad humana y realizar una amplia gama de tareas de NLP.
- **Modelado de Lenguaje**: La tarea de predecir la siguiente palabra en una secuencia basada en las palabras precedentes.

- **Procesamiento de Lenguaje Natural (NLP)**: Un campo de inteligencia artificial que se centra en permitir que las computadoras comprendan, interpreten y generen lenguaje humano.
- **Transformador**: Una arquitectura de red neuronal que utiliza mecanismos de autoatención para procesar datos secuenciales, a menudo utilizada en tareas de NLP.
- **Tokenización**: El proceso de descomponer el texto en unidades más pequeñas, llamadas tokens, como palabras, caracteres o subpalabras.
- **Validación**: El proceso de evaluar el rendimiento de un modelo entrenado utilizando conjuntos de datos de referencia y métricas.
- **Gestión de Diálogos**: El componente responsable de mantener el flujo y la coherencia de una conversación en un chatbot, incluyendo el seguimiento del historial, la gestión del contexto y la aplicación de estrategias conversacionales.
- **Modelo de Lenguaje de Propósito General**: Un modelo de lenguaje que puede realizar varias tareas de NLP sin estar diseñado específicamente para la conversación.