1. AUTOMATIZACIÓN DE ALTO VOLUMEN DE SOLICITUDES EN CUSTOMER SERVICE

CONTEXTO:

• Nuestro servicio de atención al cliente recibe un gran número de solicitudes diarias (por diversos canales: email, chat, web, etc.).

• Buscamos automatizar una parte significativa de estas interacciones mediante IA, reduciendo el tiempo de respuesta y mejorando la satisfacción del cliente.

OBJETIVO:

• Explicar tu enfoque para diseñar un sistema de automatización que permita procesar, categorizar y responder a las solicitudes.

• Proponer los pasos de un pipeline de IA (recolección de datos, limpieza, etiquetado, entrenamiento, despliegue) y cómo implementarías cada uno.

• Describir brevemente qué algoritmos o modelos de NLP (Procesamiento de Lenguaje Natural) usarías para la clasificación y/o respuesta automática (por ejemplo, modelos tipo BERT, GPT, Chatbots basados en reglas o en aprendizaje profundo, etc.).

• Mencionar las herramientas tecnológicas que considerarías (frameworks de NLP, librerías de IA conversacional, plataformas de despliegue en la nube, bases de datos, entre otros).

• Incluir estrategias para medir y mejorar continuamente la efectividad de las respuestas (métricas de precisión, recall, F1-score, feedback loop, etc.).

Propuesta para obtener el puesto de Ingeniero en IA,
presentada por Brian Buendia Sosa.

1.1. DEFINICIÓN DEL PROPOSITO DEL CHAT

Este chat busca automatizar las interacciones con los clientes para hacer más eficiente la atención al cliente. Esto implica manejar diversas situaciones, ya que se trata de un negocio de apparel, y los clientes pueden realizar muchos tipos de consultas, como preguntas sobre envíos, costos, tallas, disponibilidad de ciertos modelos, entre otras.

1.2. CANALES EN DONDE SE VA A IMPLEMENTAR

Se requiere que se puedan automatizar los distintos canales de comunicación de la empresa, desde correo electrónico y webchat, hasta redes sociales como WhatsApp, Facebook o Instagram.

1.3. SELECCIONAR LAS HERRAMIENTAS

Los modelos BERT y GPT tienen características y aplicaciones distintas que los hacen adecuados para diferentes tareas. BERT es más eficiente para tareas que requieren comprensión contextual, como el análisis de sentimientos o el reconocimiento de entidades. Sin embargo, requiere una mayor cantidad de datos de entrenamiento y un proceso de fine-tuning más extenso para alcanzar un buen rendimiento. Por otro lado, GPT es más adecuado para generar texto creativo, realizar traducciones o adaptarse a una amplia gama de tareas con un enfoque de few-shot learning. Además, GPT tiene una velocidad de inferencia más rápida en comparación con BERT, lo que lo hace más adecuado para aplicaciones en tiempo real.

Una de las diferencias más notables es que las respuestas generadas por BERT suelen ser más estáticas, es decir, el modelo no tiene tanta creatividad para generar texto, ya que se enfoca principalmente en clasificar el estilo del texto. Esto podría ser una limitación, ya que las respuestas del chatbot podrían resultar muy genéricas. Además, si el cliente formula una pregunta de manera compleja o poco clara, el modelo podría responder de forma incorrecta.

Por otro lado, los modelos GPT, al estar entrenados con una cantidad de datos mucho mayor, son más efectivos para generar texto de manera creativa. Esta característica es beneficiosa para un chatbot, ya que las respuestas se sentirían más naturales y variadas. Sin embargo, esta misma creatividad puede ser un problema si el modelo no está bien ajustado, ya que podría caer en "alucinaciones" y proporcionar respuestas incorrectas o fuera de contexto.

Si ya se cuenta con una lista de preguntas frecuentes y respuestas bien estructuradas, la mejor opción sería BERT. No obstante, si el objetivo es que el chatbot se sienta más natural y pueda interpretar de manera más efectiva las preguntas de los clientes, GPT es una opción más adecuada.

Para este proyecto, se propone utilizar GPT debido a su capacidad para responder de manera creativa y su mejor interpretación de las consultas realizadas por los clientes. En este caso el modelo GPT2 ofrece buenas características para generar respuestas creativas.

1.4. DETERMINAR LA PERSONALIDAD DEL CHATBOT

Debido a que la marca está enfocada en moda para un público joven, lo ideal sería que el chatbot tuviera un tono relajado, sin tanta formalidad, pero evitando caer en un tono infantil o irrespetuoso. Además, esta personalidad se construirá basándose en el dataset de chats o mensajes previos del equipo de comunicación de la empresa.

Se buscará entrenar al modelo con una base de datos relacionada con la empresa, incorporando toda la información relevante, como la descripción, metas, visión y otros datos importantes. Asimismo, lo ideal sería entrenarlo con un historial de chats o publicaciones previas, para que el modelo pueda aprender el tipo de preguntas o solicitudes que suelen hacer los clientes, así

como la forma adecuada de responder. Esto también permitirá que el modelo reconozca las diversas formas en que los clientes pueden formular preguntas que, en esencia, se refieren a lo mismo. A continuación, se muestra un ejemplo para clarificar este punto:

Cliente: para cuando va a llegar mi pedido?

Chat: ¡Hola! Tu pedido se encuentra en camino y se tiene planeado que va a llegar el día de mañana.

Cliente: que dia me va a llegar mi paquete?

Chat: ¡Hola! Tu pedido se encuentra en camino y se tiene planeado que va a llegar el día de mañana.

Aunque las preguntas están formuladas de manera diferente, todas se refieren a la disponibilidad de una talla específica. El modelo debe ser capaz de identificar esto y responder de manera coherente y precisa en cada caso.

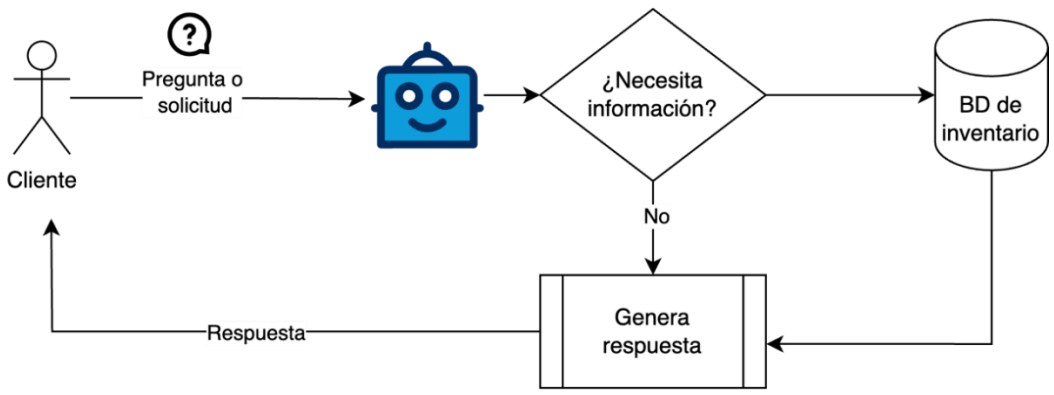
1.5. DISEÑO DEL FLUJO DE LA CONVERSACIÓN

Una vez que el modelo esté entrenado y haya adoptado el tono amable y cercano de los actuales responsables de responder las dudas, el siguiente paso es adaptarlo a las distintas situaciones a las que podría enfrentarse.

Para comenzar, es necesario realizar un análisis de todos los tipos de consultas que el chatbot deberá resolver y los escenarios a los que estará expuesto por parte de los clientes. Podemos considerar que el chatbot manejará preguntas generales sobre la empresa, pero también se enfrentará a situaciones en las que necesitará resolver dudas para las que no tiene información directamente en su red neuronal. Estas situaciones pueden incluir consultas sobre inventarios, dudas sobre modelos específicos, tallas, envíos, agendar reuniones o llamadas, entre otras. Dada la naturaleza cambiante de estos datos, no es viable entrenar al modelo con esta información, ni sería rentable actualizarlo constantemente conforme los datos cambien.

En este caso, se propone que el modelo esté conectado a una base de datos que contenga la información actualizada en tiempo real. Cada vez que se haga una pregunta que requiera acceder a estos datos, el chatbot realizará una petición a la base de datos para obtener la información necesaria.

Se sugiere que, cuando se reciba una solicitud de información relacionada con inventarios, tallas, precios o envíos, el chatbot haga una consulta a la base de datos y, con la información obtenida, genere una respuesta adecuada utilizando prompts. De esta manera, el chatbot podrá proporcionar respuestas precisas y actualizadas sin necesidad de ser reentrenado constantemente.



Una técnica consiste en proporcionar un prompt al sistema para que genere una respuesta que nos permita identificar cuándo está solicitando información específica. Por ejemplo:

Responde con las siguientes claves si te hacen una pregunta de la que no tienes informacion:
@inventario: en caso de que te soliciten informacion de inventario de una prenda en especial.
@talla: en caso que te soliciten informacion de tallas.
@precio: en caso de que te soliciten informacion del precio de una prenda.
@envio: en caso de que te soliciten informacion sobre envios.
@modelo: menciona el modelo sobre el cual quieran obtener información.

Esto puede generar cadenas de texto con claves que representen la información solicitada, las cuales se utilizarán posteriormente para buscar los datos correspondientes en la base de datos.

A continuación, se muestra un ejemplo de cómo podría funcionar este proceso:

Cliente: hay en existencia camisetas del modelo X en color azul en talla M y cuál es el precio?

En este caso, el modelo solicitará información específica sobre el modelo X, consultando la existencia de la talla M en color azul. La respuesta obtenida de la base de datos podría generar un prompt como el siguiente, el cual se proporcionará al modelo para que genere la respuesta final:

```
prompt = "Pregunta del usuario: {pregunta_usuario}
          Responde a la pregunta del usuario usando la siguiente información. Usa
          el mismo tono amable en tu respuesta.
    stock: {stock}
    precio: {precio}"

respuesta = modelo(prompt)
```

chatbot: Sí tenemos en existencia camisetas modelo X en color azul talla M ☺. El costo es de $500. Si deseas comprar esta prenda ¡visita nuestro sitio y haz tu pedido!

Al hacer que el modelo se base en información real, se puede evitar que genere respuestas fantasiosas o incorrectas. Para evaluar su capacidad de responder de manera acertada, se propone utilizar métricas como Precision, Recall, F1-Score y BertScore.

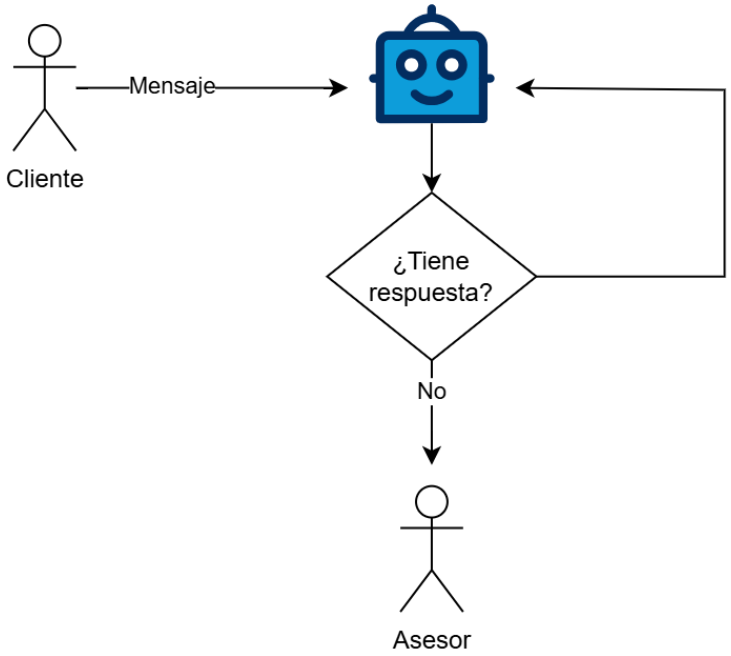
Precision: Evalúa el porcentaje de respuestas generadas por el chatbot que son correctas.

Recall: Mide el porcentaje de preguntas correctamente respondidas, de todas aquellas que el chatbot podría haber respondido correctamente.

F1-Score: Combina Precision y Recall para proporcionar una evaluación general del desempeño del chatbot.

Además, se sugiere utilizar la métrica BertScore, la cual evalúa qué tan similares son dos oraciones en términos de significado. Esto es especialmente útil en aplicaciones de chatbots, donde el modelo puede generar respuestas con diferentes palabras pero que expresen el mismo significado. Por ejemplo, las frases "El costo es de $50" y "Tiene un precio a la venta de $50" son equivalentes en significado, aunque estén formuladas de manera distinta. BertScore permite evaluar estas variaciones, centrándose en el significado más que en la forma específica de la respuesta.

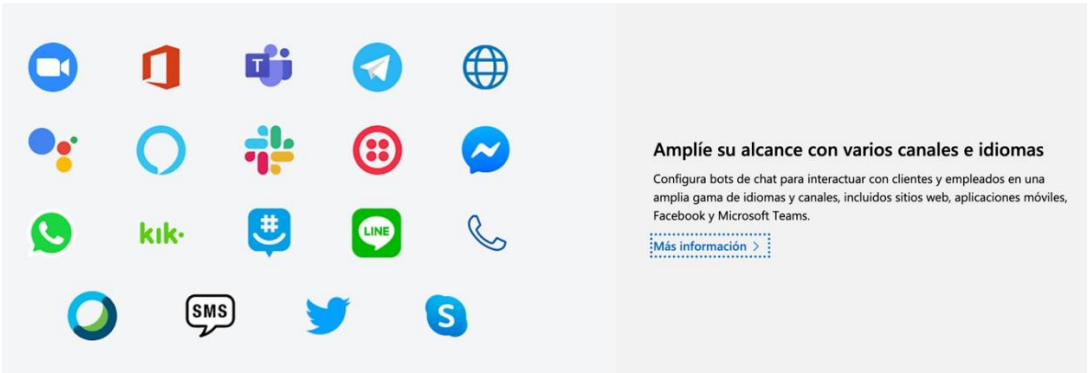
Finalmente, es importante que el chatbot cuente con la funcionalidad de transferir al cliente a un asesor humano en caso de que no tenga una respuesta adecuada o si el cliente solicita explícitamente hablar con una persona. Esto garantiza una experiencia de usuario más completa y satisfactoria.



1.6. INTEGRACIÓN Y PRUEBAS

Dada la naturaleza de esta aplicación, lo más recomendable sería utilizar un servicio en la nube para desplegar el modelo. Esto se debe a que el chatbot debe estar disponible las 24 horas del día, ya que no se puede predecir en qué momento un cliente realizará una solicitud de información. Además, algunos servicios en la nube facilitan la integración con otras herramientas, como correo electrónico o plataformas de mensajería.

Entre las opciones más destacadas se encuentran AWS, Azure y GCP. Para este caso, se propone utilizar Azure Bot Services, ya que permite desplegar el modelo previamente entrenado de manera eficiente. Una de sus principales ventajas es su conectividad sencilla con múltiples aplicaciones de chat, como WhatsApp, Telegram o Facebook, así como su integración con servicios de correo electrónico. Esto lo convierte en una opción ideal para garantizar que el chatbot esté accesible en los canales más utilizados por los clientes y pueda manejar consultas de manera ágil y efectiva.



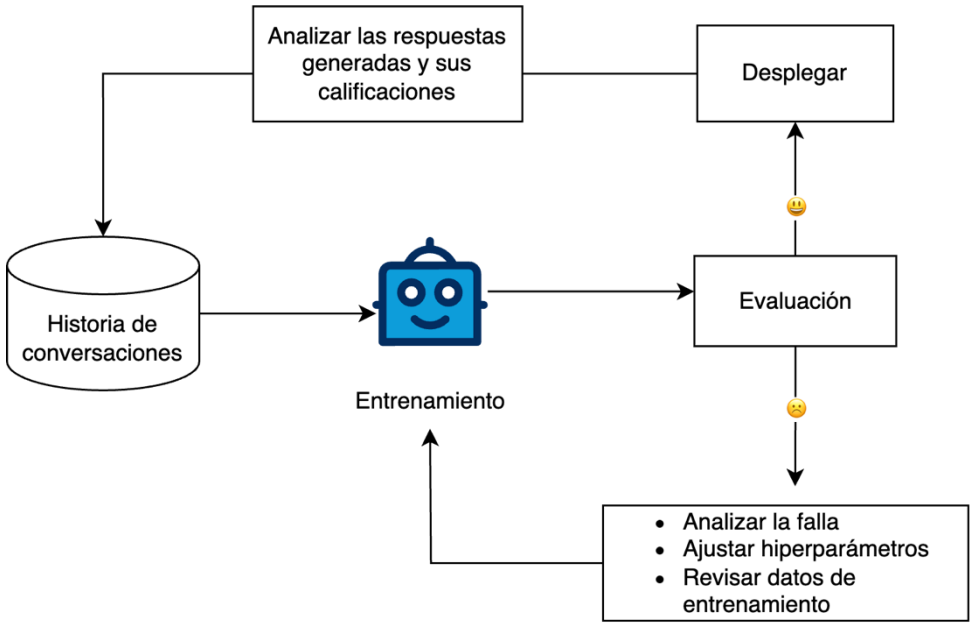
1.7. LANZAMIENTO Y MONITOREO

Es fundamental monitorear constantemente el modelo para asegurarse de que funcione correctamente y no proporcione información errónea o que no resuelva las dudas de los clientes. Además, es importante considerar que, con el tiempo, los estilos, los requerimientos de los clientes e incluso su forma de expresarse pueden cambiar, por lo que es necesario mantener un aprendizaje continuo del modelo.

Una buena práctica es implementar una opción para que los clientes evalúen la ayuda del chatbot después de usarlo. Esto puede hacerse mediante una calificación simple, como un sistema de estrellas (del 1 al 5 ⭐) o con tres niveles de emoticones (😊, 😐, 😞).

Todas las conversaciones pueden almacenarse y clasificarse según la calificación recibida. Las respuestas con las mejores calificaciones pueden utilizarse para seguir entrenando al modelo mediante aprendizaje continuo en intervalos regulares. Por otro lado, las respuestas con malas calificaciones pueden ser revisadas por personal humano, quien identificará en qué tipo de preguntas o situaciones está fallando el chatbot. Esta información permitirá corregir manualmente las fallas y luego entrenar al sistema con los datos ajustados, mejorando así su precisión y eficacia con el tiempo.

Este enfoque garantiza que el chatbot evolucione y se adapte a las necesidades cambiantes de los clientes, manteniendo un alto nivel de satisfacción y confiabilidad.



Para determinar cada cuánto se debe reentrenar el modelo utilizando la técnica de fine-tuning, es importante considerar la cantidad de solicitudes que se reciben y planificar las fechas de reentrenamiento en función de este volumen. Por ejemplo, si en un período de 15 días se generan alrededor de 1000 solicitudes, este sería un momento adecuado para planificar el reentrenamiento.