Automatización de Procesos de Cualificación en IQAP mediante chatbot de IA para Ingreso a Universidades Internacionales

INFORME DE TESIS

Presentado para obtener el título de

MAGÍSTER EN MATEMÁTICAS APLICADAS

Y CIENCIAS DE LA COMPUTACIÓN

Jairo Vladimir Tamayo Ramírez

Rubén Darío Rico González

Gustavo Adolfo Noriega Cárcamo

Dirección:

Édgar José Andrade Lotero

Universidad del Rosario

Escuela de Ingeniería, Ciencia y Tecnología

Magíster en Matemáticas Aplicadas y Ciencias de la Computación.

DEDICATORIA

A nuestras familias y mascotas

“The Talking Robot was designed to answer questions, and only such questions as it could answer had ever been put to it.”

Asimov, I. Robbie, 1968

RESUMEN

Uno de los grandes desafíos empresariales es poder interactuar con sus potenciales usuarios de una manera cómoda y fluida, el presente proyecto ayudó a la empresa SOFIRI PTY LTD a mejorar su Plataforma de Cualificación Instantánea de Aspirantes IQAP con la implementación de técnicas avanzadas de Procesamiento del Lenguaje Natural (NLP) y modelos de lenguaje de gran escala (LLMs) de compañías líderes en AI. Como resultado de la aplicación de esta propuesta se logró desarrollar un chatbot que consiguió mejores resultados que el actual e interactuó con los usuarios de forma más natural; resultados que se midieron mediante la aplicación de una encuesta a los usuarios de prueba para evaluarlo de forma cualitativa y el cálculo de métricas para hacerlo cuantitativamente.

ABSTRACT

One of the major business challenges is to interact with potential users in a comfortable and fluid manner. This project helped the company SOFIRI PTY LTD to improve its Instant Qualification of Applicants Platform (IQAP) by implementing advanced Natural Language Processing (NLP) techniques and large-scale language models (LLMs) from leading AI companies. As a result of implementing this proposal, a chatbot was developed that achieved better results than the existing one and interacted with users in a more natural way; results that were measured by conducting a survey of test users to evaluate it qualitatively and calculating metrics for quantitative assessment.

Palabras clave: Automatización de procesos, chatbot, Inteligencia artificial, Procesamiento de Lenguaje Natural (NLP), experiencia del usuario, Transformers, LLM.

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

ABREVIATURAS

NLP: Natural Language Processing (Procesamiento del Lenguaje Natural).

LLM: Large Language Models (Modelos de Lenguaje de Gran Escala).

IQAP: Instant Qualified Applicant Platform.

GUS: Generalized User Simulation (Simulación Generalizada del Usuario).

PT: Pty Ltd (Proprietary Limited).

BLEU: Bilingual Evaluation Understudy.

ROUGE: Recall-Oriented Understudy for Gisting Evaluation.

T5: Text-to-Text Transfer Transformer (Modelo de Transformadores de Transferencia de Texto a Texto).

CAPÍTULO 1. INTRODUCCIÓN

Este informe presenta la optimización de la Plataforma de Cualificación de Aspirantes (IQAP Instant Qualified Applicant Platform) de SOFIRI PTY LTD, una empresa con sede en Sídney, Australia, que utiliza el software IQAP para gestionar la inscripción y admisión a instituciones de educación internacional. IQAP es una herramienta de SOFIRI para automatizar la recopilación de datos de aspirantes a programas educativos internacionales para su cualificación. La optimización se enfoca en el desarrollo de un chatbot que emplea Procesamiento del Lenguaje Natural (NLP) y Modelos de Lenguaje de Gran Escala (LLMs). A través de la implementación de tecnologías avanzadas de NLP, el nuevo chatbot captura la información y comprende el lenguaje natural de los usuarios, permitiendo generar preguntas adaptativas que se ajustan en tiempo real a las respuestas de los usuarios. Esta funcionalidad mejora la precisión en la captura de datos necesarios para la cualificación de aspirantes y optimiza el flujo de interacción.

La necesidad de este desarrollo surgió del análisis del sistema anterior, que registra una tasa de abandono superior al 45% durante las etapas preliminares de cualificación. Este sistema se basaba en un conjunto fijo de preguntas y no se adaptaba a las variaciones en las respuestas de los usuarios, resultando en una experiencia rígida y a menudo frustrante.

En las pruebas piloto realizadas, se aplicó una métrica de similitud normalizada para evaluar la captura de las respuestas de los usuarios a través del chatbot, alcanzando un 98.79% de similitud en las respuestas, lo que indica una alta precisión en la captura de datos requeridos. La eficiencia del modelo GPT-4o, utilizado en el chatbot, fue destacada, mostrando una mejor interpretación y procesamiento de las entradas de los usuarios comparado con otros modelos evaluados. Además, las encuestas realizadas a los participantes reflejaron una percepción positiva del chatbot. Más del 57% de los usuarios reportaron una interacción más fácil con este chatbot en comparación con otros, y el 78% calificó la fluidez de la interacción como razonablemente fluida o muy fluida.

En comparación con el chatbot preexistente, el nuevo chatbot propone una redacción de preguntas más fluida y natural, simulando una conversación entre el usuario y un asesor de educación. Esta

mejora se refleja en la Tabla 1, que compara las preguntas del chatbot anterior con las generadas por el chatbot mejorado.

Tabla 1. Preguntas del chatbot anterior versus la generación de preguntas por el chatbot mejorado

Chatbot anterior	Chatbot mejorado
What city do you live in?	Hi Vlad, could you please share which city you currently reside in?
What was your latest level of study? Just choose one of the options I am listing.	What country are you considering for your studies, Tony?
In what country would you like to study?	What country are you considering for your studies abroad?
I want to make sure we find you something that fits your budget. How much do you expect to pay in course fees per year?	Understanding your financial situation is crucial for finding the right study options. Can you give me an idea of your budget range for yearly tuition fees for your intended course in Australia?
What is your surname?	May I ask for your last name, Vladimir?

Este documento detalla el proceso técnico de integración de las tecnologías de NLP, las metodologías empleadas para el diseño y la implementación del chatbot, y los resultados significativos derivados de esta innovación tecnológica en la operación de la plataforma IQAP.

CAPÍTULO 2. JUSTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA

Planteamiento del Problema

Este proyecto responde a una necesidad de SOFIRI PTY LTD, una empresa con sede en Sídney, Australia, que utiliza el software IQAP para gestionar la inscripción y admisión a instituciones de educación internacional. El objetivo principal es mejorar la interacción con los usuarios a través de la actualización del chatbot existente, optimizando así la cualificación de los estudiantes interesados en programas educativos en el extranjero.

El chatbot actual se basa en un conjunto de preguntas predefinidas para recopilar información de los postulantes. Estas preguntas son fundamentales para precalificar a los candidatos y tomar decisiones de admisión informadas. Abarcan detalles como el nombre del estudiante, país de origen, fecha deseada de inicio del programa, preferencias de destino, así como detalles económicos y disponibilidad de recursos, entre otros.

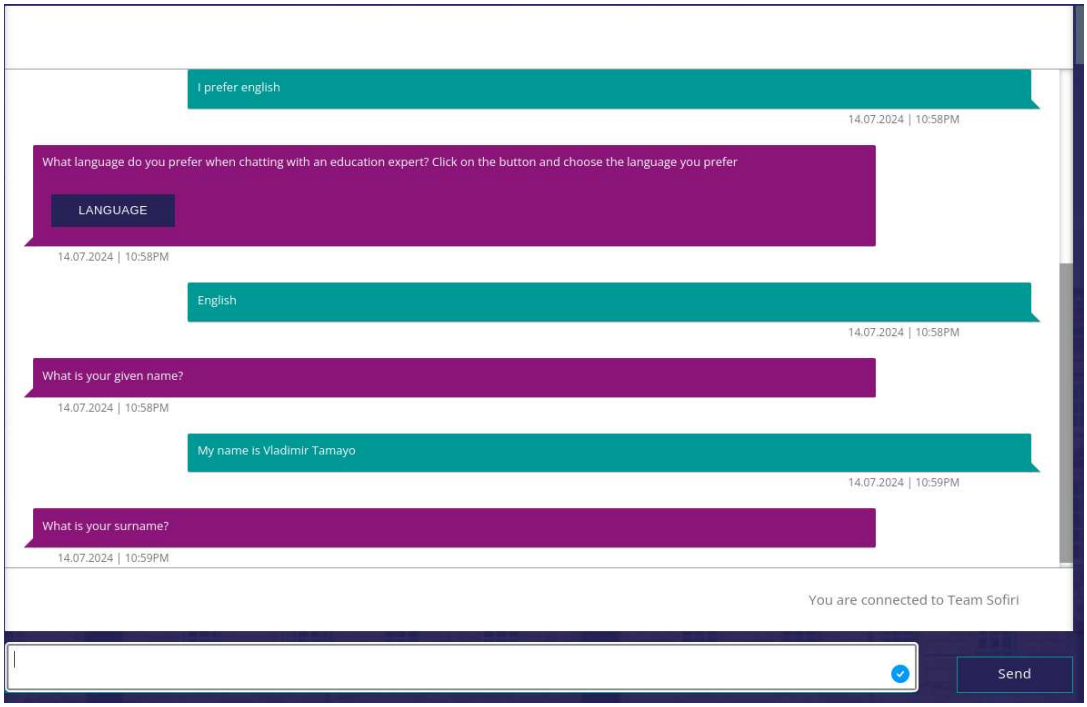

Imagen 1. Ejemplo del Chatbot anterior, en donde solicita información ya suministrada por el usuario.

Se ha recopilado información histórica de registros de usuarios desde 2018, acumulando un total de más de 500,000 registros a través del chatbot existente en plataformas como Facebook Messenger, aplicaciones Android e iOS, y directamente en el sitio web de SOFIRI PTY LTD. A partir de estos datos, se observó que más de 200,000 usuarios, representando el 46% del total, abandonaron el proceso de registro sin completar la información solicitada por el chatbot, necesaria para que el software realice la cualificación inicial.

El problema central abordado en esta tesis de grado es la necesidad de mejorar la interacción con los usuarios a través de la actualización del chatbot existente. Con una interacción más fluida se espera aumentar la tasa de finalización de la fase de cualificación de los solicitantes. Además, se busca asegurar una mayor precisión en la obtención de la información de los usuarios. El chatbot actual, basado en listas de verificación, no proporciona una experiencia interactiva y conversacional, lo que lleva al abandono por parte de los usuarios antes de completar el proceso de cualificación. También se han observado registros completados con errores en los datos proporcionados, lo que limita la exactitud y utilidad del sistema.

Esta tesis de maestría busca abordar estos problemas mediante la aplicación de técnicas de NLP para rediseñar el chatbot. El objetivo principal es mejorar la interacción con los usuarios, garantizando la precisión en la recopilación de datos y una mejora sustancial en la formulación de preguntas, lo que mejorará la inscripción de aspirantes a programas universitarios internacionales. Al proporcionar una experiencia de usuario más agradable y reducir las tasas de abandono, se espera aumentar la eficacia general del proceso de cualificación y mejorar la percepción del chatbot del software IQAP.

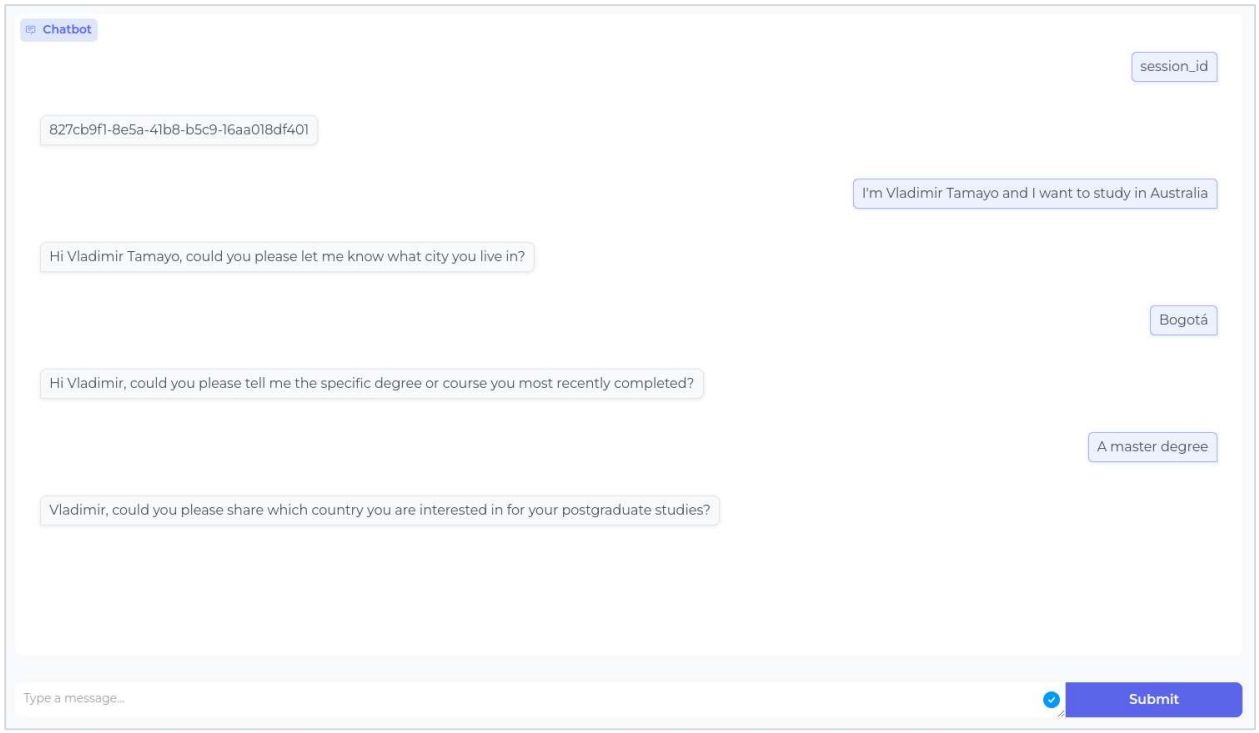

Imagen 2. Chatbot mejorado desarrollado en el trabajo de grado. No solicita información previamente ingresada por el usuario.

Justificación

Este proyecto surge de la necesidad de abordar un problema concreto en el proceso de registro para la cualificación de aspirantes en el entorno del software IQAP. Aporta conocimientos sobre la implementación de arquitecturas, despliegue en entornos de producción, algoritmos y métodos de NLP aplicados a la implementación de chatbots en contextos reales.

La integración de técnicas avanzadas de NLP, como el uso de Transformers y grandes modelos de lenguaje (LLMs), para crear un chatbot más interactivo y preciso, establece un ejemplo para futuros desarrollos en este ámbito. Los resultados y las lecciones aprendidas de este proyecto contribuyen al conocimiento en la selección de métodos de NLP para la implementación de chatbots, no solo para la cualificación de aspirantes a estudio en universidades internacionales, sino también en diversos escenarios reales. Esto promueve el desarrollo de soluciones de IA que mejoran la interacción humano-máquina en una amplia gama de aplicaciones.

Con una interacción más fluida, se espera aumentar la tasa de finalización de la fase de cualificación de los solicitantes. Además, mejorar la precisión en la recolección de datos no solo optimiza el proceso de cualificación de estudiantes, sino que también puede aumentar el número de aspirantes admitidos o que avancen a las siguientes etapas del proceso.

Desde una perspectiva social, aunque el objetivo principal del chatbot es mejorar los procesos de cualificación, se espera que indirectamente facilite el acceso a la educación superior internacional. Un sistema de cualificación más preciso puede, sin pretenderlo específicamente, atender a una mayor diversidad de estudiantes, eliminando barreras idiomáticas y de comprensión que podrían impedir la inscripción de candidatos calificados. Este efecto colateral contribuiría a promover la inclusión y la equidad en el acceso a oportunidades educativas, alineándose con el propósito de SOFIRI de facilitar la conexión entre aspirantes de distintos países y las instituciones educativas.

El uso de técnicas avanzadas de NLP y LLMs en la mejora del chatbot contribuye a la gestión del software de SOFIRI. Además, los conocimientos generados pueden aplicarse a otros sectores que requieren soluciones de interacción humano-máquina, como el seguimiento de procesos industriales y empresariales en la cadena de abastecimiento, recursos humanos, manuales de mantenimiento y servicio al cliente, entre otros servicios. Estas aplicaciones resultan valiosas en contextos donde las compañías enfrentan dificultades para destinar recursos suficientes a la atención en tiempo real de los usuarios.

En resumen, este proyecto no solo aborda una necesidad técnica específica en el entorno del software IQAP, sino que también tiene amplias implicaciones económicas y sociales. Al proporcionar una experiencia de usuario más agradable y reducir las tasas de abandono, se espera aumentar la eficacia general del proceso de cualificación y mejorar la percepción del software IQAP.

CAPÍTULO 3. OBJETIVOS

Objetivo general

Desarrollar e implementar un chatbot basado en inteligencia artificial en un entorno de pruebas equivalente al software IQAP, utilizando técnicas recientes de NLP y aprendizaje automático, simulando la interacción en idioma inglés con un consejero humano en educación, con el propósito de mejorar la comunicación con los usuarios y facilitar la finalización del proceso de cualificación de aspirantes internacionales.

Objetivos específicos

- Evaluar la precisión en la extracción de información del chatbot mediante la comparación de los datos obtenidos con datos históricos, a partir de un conjunto de requerimientos de información o “slots” previamente definidos.

- Ejecutar pruebas piloto del chatbot con un grupo experimental para evaluar su funcionamiento, identificar errores y obtener retroalimentación para formular iniciativas de mejora.

- Desarrollar un chatbot en idioma inglés, integrable en el entorno de pruebas del software IQAP para llevar a cabo la cualificación de aspirantes y evaluar el desempeño en términos de precisión en la extracción de información y calidad de la interacción.

CAPÍTULO 4. MARCO TEÓRICO

Este marco teórico aborda los conceptos clave que apoyan el desarrollo del proyecto, que consiste en un chatbot inteligente para automatizar la cualificación de usuarios. Aquí, se explicarán las ideas esenciales sobre el procesamiento de lenguaje natural y los modelos de lenguaje. Este enfoque se adopta porque el proyecto se centra principalmente en la aplicación de estas tecnologías para mejorar la plataforma IQAP y la experiencia del usuario.

En este capítulo se examinarán las arquitecturas de los transformers [1] y su impacto en la comprensión y generación de lenguaje natural. Además, se explorarán técnicas de prompting y aprendizaje en contexto, y se describirá la evolución y funcionamiento de los chatbots basados en reglas y corpus. Finalmente, se discutirá la evaluación de estos sistemas utilizando métricas como BLEU, ROUGE, BERTScore y Levenshtein, destacando su relevancia para medir la precisión y efectividad en la generación de texto.

El NLP es una subdisciplina de la inteligencia artificial que se enfoca en la interacción entre las computadoras y el lenguaje humano. Su objetivo es dotar a las máquinas de la capacidad para comprender, interpretar y generar lenguaje humano de manera precisa y eficiente. El NLP se utiliza en una amplia gama de aplicaciones, desde la traducción automática hasta el análisis de sentimientos y la generación de texto, transformando la manera en que interactuamos con la tecnología [2, p. 29].

Modelos de lenguaje: Fundamentos y Avances

Los modelos de lenguaje (Language Models, LMs) son modelos de inteligencia artificial que asignan una probabilidad a una secuencia de palabras. Este proceso se basa en el aprendizaje de patrones y estructuras del lenguaje a partir de grandes corpus de texto. Los LLMs son fundamentales en muchas aplicaciones de NLP, como la traducción, el análisis de sentimientos y la generación de texto. Existen diferentes tipos de modelos de lenguaje, como los modelos N-gram, que utilizan la probabilidad de una palabra basada en las n-1 palabras anteriores y los modelos basados en redes neuronales, que superan las limitaciones de los N-gram [2, pp. 31-35] al usar

embeddings y redes neuronales recurrentes (RNNs) o transformers para capturar dependencias a largo plazo en el texto [2, pp. 185-186].

Modelos N-gram y sus Limitaciones

Los modelos N-gram fueron una de las primeras aproximaciones en el NLP. Calculan la probabilidad de una palabra basada en las n-1 palabras anteriores. Formalmente, la probabilidad de una secuencia de palabras $w_1, w_2, \ldots, w_T$ en un modelo N-gram se define como:

$$P(w_1, w_2, \ldots, w_T) = \prod_{i=1}^{T} P\left(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1}\right)$$

Sin embargo, estos modelos tienen una capacidad limitada para capturar dependencias a largo plazo debido a su enfoque en contextos locales. Además, algunos problemas de ausencia de datos, lo que significa que muchos posibles n-grams no están presentes en el corpus de entrenamiento, resultando en probabilidades cero para estos n-grams en el conjunto de prueba. Esto puede llevar a una subestimación de la probabilidad de ciertos eventos y afectar negativamente el rendimiento del modelo en aplicaciones prácticas [2, p. 43].

Redes Neuronales Recurrentes (RNNs)

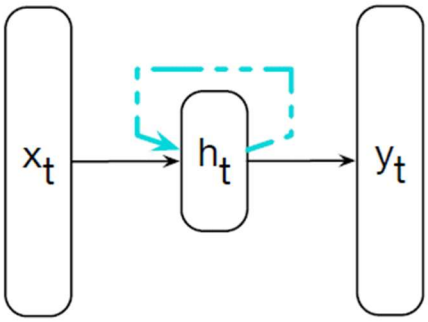

Figura 1. Red neuronal recurrente simple según Elman (1990) [5, p. 186]

Las RNNs pueden mantener información sobre las dependencias temporales en secuencias de datos. Matemáticamente, el estado oculto h_t de una RNN en el tiempo t se define como:

$$h_t = f(W_h h_{t-1} + W_x x_t + b)$$

donde W_h y W_x son matrices de pesos, b es el sesgo y f es una función de activación no lineal, como tanh o ReLU [6]. Sin embargo, las RNNs enfrentan problemas con el desvanecimiento del gradiente, lo que dificulta su efectividad en secuencias largas [1].

Embeddings

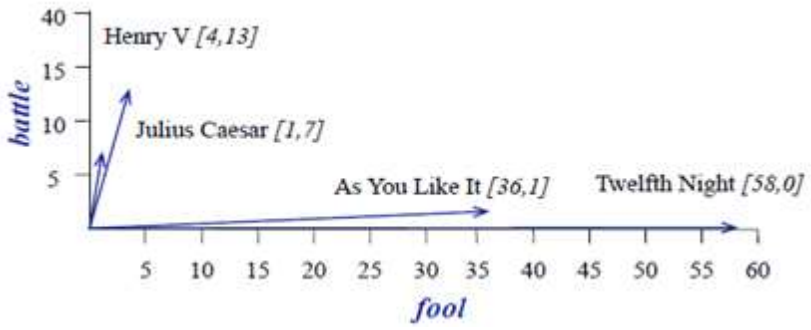

Figura 2. Representación vectorial de texto

Nota: recuperado de [2, p. 110]

En la figura anterior se presenta la definición gráfica de qué es un embedding, se ilustran cuatro representaciones vectoriales de sendas frases y la evaluación de su precisión, los embeddings representan una palabra como un punto en un espacio semántico multidimensional que es derivado de la distribución de las palabras vecinas, pueden representar palabras, frases o documentos, que capturan la semántica de los textos y permiten medir similitudes de manera eficiente.

Transformers: Una Revolución en el NLP

La introducción de la arquitectura de transformers [1] marcó un cambio significativo en el campo del NLP. Los transformers utilizan mecanismos de atención para procesar secuencias de datos de manera más efectiva que las RNNs, permitiendo una mejor captura de dependencias a largo plazo y una mayor paralelización en el procesamiento de datos. Esto mejora significativamente la eficiencia tanto en el entrenamiento como en la inferencia de modelos de lenguaje.

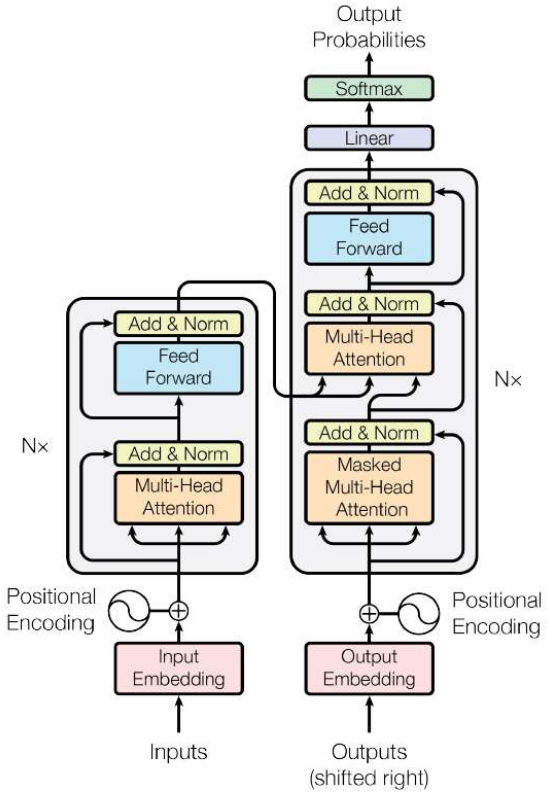

Figura 3. Transformers - Arquitectura del Modelo [1].

Como se observa en la figura anterior [1], la arquitectura del modelo transformer se compone de una serie de bloques de codificación y decodificación (encoder – decoder), que utilizan mecanismos de autoatención y atención multi-cabezal para manejar dependencias a largo plazo en las secuencias de entrada. Esto permite que los transformers procesen datos en paralelo, lo que mejora significativamente la eficiencia del entrenamiento y la inferencia [2, p. 43].

Mecanismos de Atención en Transformers

Los Transformers representan una arquitectura que elimina el uso de recurrencias y convoluciones, confiando exclusivamente en mecanismos de atención para captar dependencias globales entre las entradas y las salidas. Esta estructura permite la paralelización y reduce los tiempos de entrenamiento, ofreciendo ventajas considerables sobre arquitecturas tradicionales [2, p. 212]

Atención por Producto Escalar (Scaled Dot-Product Attention)

El mecanismo básico de atención, denominado Atención por Producto Escalar, se ilustra en la Figura 4, lado izquierdo. Este proceso comienza con la generación de tres conjuntos de vectores a partir de la entrada: las consultas (Q), las claves (K) y los valores (V). Estos vectores se derivan multiplicando la entrada del modelo por matrices de pesos entrenables específicas para consultas, claves y valores: [1]

Consultas (Q): $Q = W^{Q} \cdot x$, representan lo que se busca entender o destacar de la entrada.

Claves (K): $K = W^{K} \cdot x$ actúan como índices para recuperar la información relevante.

Valores (V): $V = W^{V} \cdot x$, contienen los datos que serán ponderados y resumidos basándose en la atención calculada.

Aquí, W^{Q}, W^{K}, y W^{V} son matrices de pesos específicas, y x es el vector de embedding de la palabra. Estos vectores se organizan en matrices Q, K, y V cuando se calcula la atención sobre múltiples consultas simultáneamente, lo que lleva a la siguiente operación matemática:

$$Attention\ (Q, K, V) = softmax\ \left(\frac{QK^{T}}{\sqrt{d_k}}\right)V$$

Este cálculo efectúa el producto escalar entre cada consulta y todas las claves, lo escala, y aplica una función softmax para obtener pesos que son utilizados para ponderar los valores. Este tipo de atención es importante para manejar relaciones dentro de secuencias largas.

La puntuación de atención α_{ij}, que define cuánto enfoque se le da a cada parte de los valores, se calcula como:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{ik})}$$

donde $e_{ij} = \frac{(h_i W_q)\,(h_i W_k)^T}{\sqrt{d_k}}$ representa la relevancia escalada entre las representaciones de consulta y clave, y d_k es la dimensión de las claves [2, pp. 214-216].

Atención Multi-Cabezal

La **Atención Multi-Cabezal**, ilustrada en la Figura 4, lado derecho, extiende el mecanismo de atención simple permitiendo que el modelo procese la entrada a través de múltiples "cabezas" de atención en paralelo. Esto posibilita que el modelo capture información desde varios subespacios representativos de manera simultánea. Cada cabeza produce su propia salida, que luego se concatenan y proyectan a través de una capa lineal final, enriqueciendo la capacidad del modelo para integrar y entender diversos aspectos de la entrada.

La siguiente figura ilustra el concepto de atención multi-cabezal, destacando cómo se procesan múltiples secuencias de atención en paralelo.

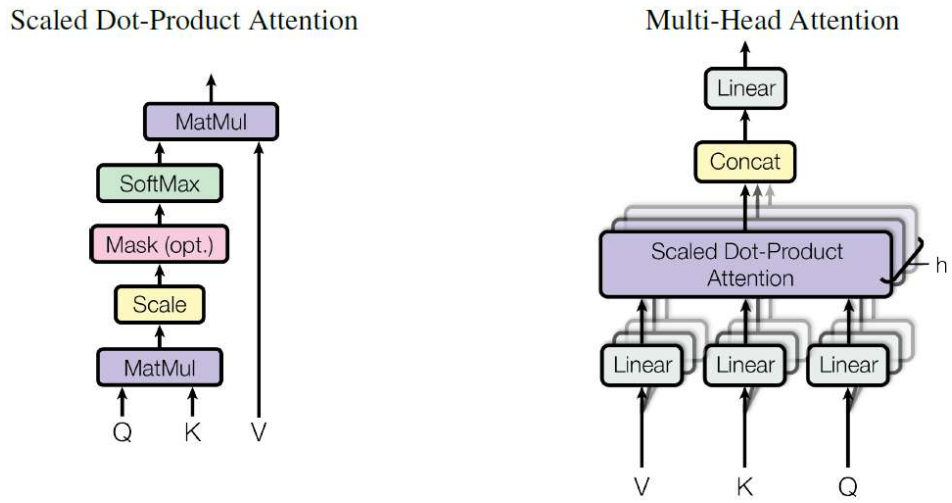

Figura 4. (izquierda) Atención por Producto Escalar Normalizado. (derecha) Atención Multi-Cabezal [1]

Componentes Adicionales de los Transformers

Adicionalmente, cada capa en el codificador y decodificador (encoder – decoder) contiene una red neuronal feed-forward totalmente conectada, que se aplica a cada posición de la secuencia de

manera independiente y con los mismos parámetros. Esta red consta de dos transformaciones lineales con una activación ReLU intermedia [2, p. 203].

Codificaciones Posicionales en Modelos Transformer

Dado que los transformers no utilizan convoluciones ni recurrencias, es importante inyectar información sobre el orden de la secuencia para que el modelo pueda interpretar correctamente la secuencia de entrada. Esto se logra añadiendo codificaciones posicionales a los embeddings de entrada. Estas codificaciones proporcionan al modelo datos sobre las posiciones relativas y absolutas de los tokens dentro de la secuencia. Las codificaciones posicionales pueden ser de naturaleza fija o aprendidas durante el entrenamiento, y se suman a los embeddings de entrada en las pilas (stacks) del codificador y del decodificador [2, pp. 218-219].

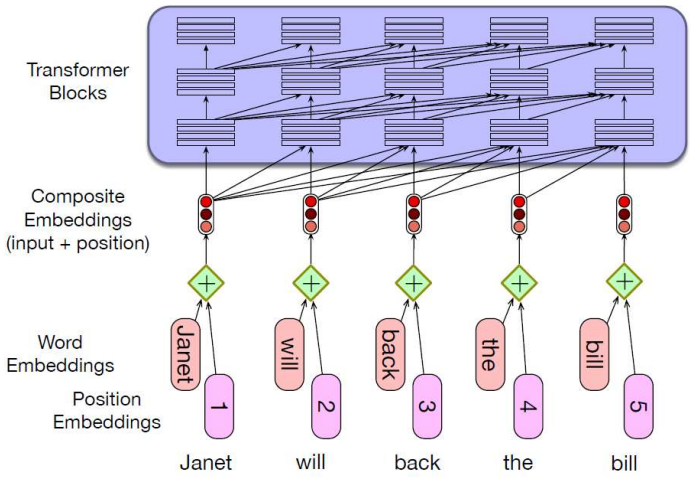

Figura 5. Modelado simple de posición: combinar incrustaciones de palabras y posición

La adición de 'codificaciones posicionales' a los embeddings de entrada es fundamental, especialmente en la base de las capas del codificador y del decodificador. Estas codificaciones tienen la misma dimensión dmodel que los embeddings, permitiendo que ambos conjuntos de datos se sumen directamente. Existen diversas estrategias para implementar estas codificaciones, tanto aprendidas como fijas. [1, p. 6].

En el trabajo "Attention Is All You Need", se utilizan funciones seno y coseno de diferentes frecuencias para generar codificaciones posicionales. Las codificaciones para una posición pos y una dimensión i en el vector de codificación se calculan de la siguiente manera:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right)$$

donde pos es la posición del token en la secuencia, i es la posición en el vector de codificación, y d_{model} es la dimensión del embedding del modelo. Esta fórmula permite que cada dimensión del vector de codificación posicional corresponda a una sinusoidal de diferente frecuencia. La base 10000 y la potencia $2_i/d_{model}$ controlan la variación de las ondas sinusoidales, facilitando que el modelo aprenda a reconocer y utilizar la información posicional [1, p. 6].

Una vez calculadas, las codificaciones posicionales se suman directamente a los embeddings de los tokens antes de ingresar a las capas del Transformer. Esto se expresa como:

$$E(x) = E_{token}(x) + PE(pos)$$

donde E(x) es el embedding final que se introduce en el Transformer, E_{token} (x) es el embedding del token x, y PE (pos) es la codificación posicional para la posición pos del token.

En conclusión, los Transformers, gracias a su arquitectura y la incorporación de codificaciones posicionales, ofrecen ventajas como la paralelización, la captura de dependencias a largo plazo y la escalabilidad. Esto los convierte en herramientas poderosas para una amplia gama de aplicaciones de procesamiento del lenguaje natural.

*Paralelización***:** A diferencia de las redes neuronales recurrentes (RNNs) y las redes neuronales convolucionales (CNNs), los transformers permiten la paralelización completa del entrenamiento y la inferencia. Esto se debe a que la autoatención y las capas feed-forward permiten que cada elemento de la secuencia sea procesado de manera independiente.

*Captura de Dependencias a Largo Plazo***:** Los mecanismos de autoatención permiten a los transformers capturar dependencias a largo plazo en las secuencias de entrada de manera más efectiva que las RNNs, que suelen tener dificultades con dependencias largas debido a problemas de desvanecimiento del gradiente.

Escalabilidad: Los transformers pueden escalarse fácilmente para entrenarse en grandes volúmenes de datos, lo que los hace adecuados para modelos de lenguaje a gran escala como BERT y GPT. Su capacidad para manejar grandes cantidades de datos y entrenarse de manera eficiente ha llevado a avances significativos en tareas de NLP como la traducción automática, el resumen de texto y la respuesta a preguntas.

En la actualidad los transformers se han convertido en la arquitectura base para muchos modelos de lenguaje preentrenados, como BERT, GPT, T5 entre otros. Estos modelos utilizan grandes corpus de datos de texto para preentrenarse y aprender representaciones profundas del lenguaje, que una vez preentrenados es posible ajustarlos (fine-tuned) para tareas específicas con conjuntos de datos más pequeños, aprovechando el conocimiento general adquirido durante el preentrenamiento [2, p. 231].

Dado el impacto y la eficacia de los transformers, es importante destacar algunos modelos desarrollados sobre esta arquitectura, como FLAN-T5 (Text-to-Text Transfer Transformer). Este modelo de lenguaje, desarrollado por Google, utiliza una arquitectura encoder-decoder que convierte todas las tareas de NLP en problemas de traducción de texto a texto, permitiendo un enfoque unificado para el entrenamiento y la inferencia. La arquitectura T5 consta de un codificador que procesa la entrada y un decodificador que genera la salida, lo que permite al modelo manejar tareas como la traducción, el resumen y la clasificación de texto.

Los modelos autorregresivos, como GPT (Generative Pre-trained Transformer), Llama, Claude y Falcon, generan texto prediciendo la siguiente palabra en una secuencia dada su historia. Estos modelos utilizan solo el decodificador (decoder) de la arquitectura de transformer y son muy efectivos para tareas de generación de texto. GPT, por ejemplo, presenta una gran capacidad para generar texto coherente y relevante en varios contextos, desde escribir artículos hasta mantener conversaciones naturales.

BERT (Bidirectional Encoder Representations from Transformers) y ROBERTA (A Robustly Optimized BERT Pretraining Approach) son modelos de lenguaje que utilizan una arquitectura de codificación bidireccional. Estos modelos se entrenan para predecir palabras enmascaradas dentro de una secuencia, lo que les permite capturar mejor el contexto bidireccional de las palabras. Los embeddings generados por estos modelos son representaciones densas que encapsulan el significado contextual de las palabras, lo que los hace muy eficientes para tareas de clasificación y recuperación de información.

Aprendizaje en Contexto y Prompts

Aprendizaje en Contexto (In-Context Learning)

El aprendizaje en contexto permite a los modelos de lenguaje adaptarse a nuevas tareas utilizando ejemplos específicos proporcionados en el contexto de entrada. Esto facilita la adaptación de los modelos a diversas tareas sin necesidad de un entrenamiento extensivo adicional, mejorando la flexibilidad y aplicabilidad de los modelos en diferentes escenarios [9].

El **prompting** relativo a los chatbots involucra suministrar entradas específicas para guiar al modelo de lenguaje hacia la generación de las salidas deseadas. Esta técnica aprovecha las capacidades de los modelos como los transformers para comprender y responder a diversos tipos de entradas, permitiéndoles realizar una gran variedad de tareas [2, p. 311].

Existe una amplia variedad de prompts que de acuerdo con la aplicación de cada solución de NLP, pueden tener diferentes grados de utilidad. La calidad de cada prompt, según su finalidad, está

directamente relacionada al desempeño de la solución. A continuación, se detallan los tipos de prompts [9, pp. 6-7]:

1. **Zero-shot Prompts**: Estos prompts consisten en dar al modelo una tarea o pregunta sin ningún ejemplo. El modelo se basa únicamente en su conocimiento preexistente para generar una respuesta, evaluando su capacidad para generalizar a partir de las instrucciones proporcionadas.

2. **Few-shot Prompts**: En este método, el prompt incluye uno o más ejemplos del output deseado. El modelo utiliza estos ejemplos para comprender mejor el contexto y el formato de la respuesta esperada. El prompting few-shot es utilizado para generar salidas más coherentes y dentro de un contexto.

3. **Prompts Estáticos**: Estos son pre escritos y no cambian. Proporcionan respuestas consistentes, asegurando que el chatbot mantenga un conjunto estable de comportamientos y respuestas. Este tipo puede funcionar muy bien para incorporar directrices morales o éticas específicas dentro del chatbot.

4. **Prompts Compuestos**: Parcialmente estáticos con secciones que pueden ser dinámicamente pobladas según la entrada. Esto permite que el modelo complete detalles específicos mientras mantiene una estructura consistente. Por ejemplo, un prompt compuesto puede incluir espacios reservados para datos específicos del usuario que se rellenan durante la conversación.

5. **Prompts Dinámicos**: Estos prompts se generan en tiempo real basándose en interacciones previas o salidas anteriores. Permiten que el chatbot se adapte a las conversaciones en curso utilizando el contexto proporcionado en prompts y respuestas anteriores. Este encadenamiento de prompts mejora la capacidad del modelo para manejar diálogos complejos y en evolución.

El **Instruction Tuning** implica realizar el ajuste fino de los modelos con un conjunto de instrucciones y salidas correspondientes. Este proceso mejora la capacidad del modelo para seguir

instrucciones complejas con precisión. Al entrenar el modelo en un conjunto diverso de tareas con instrucciones explícitas, se mejora su comprensión y ejecución de los comandos proporcionados por los usuarios.

Detalles del Few-Shot Prompting

El prompting few-shot implica incluir un pequeño número de ejemplos de tarea en el prompt de entrada para guiar al modelo. Este método explota la capacidad del modelo para reconocer patrones y estructuras en los ejemplos para producir salidas similares para nuevas entradas no vistas [2, p. 246]. El proceso del prompting few-shot se puede desglosar de la siguiente manera:

1. **Selección de Ejemplos**: Elegir ejemplos que representen el formato y contexto de salida deseados. La calidad y relevancia de estos ejemplos impactan directamente en el desempeño del modelo.

2. **Construcción del Prompt**: Construir el prompt incorporando los ejemplos dentro del texto de entrada. Asegurarse de que los ejemplos sean claros y representativos de la tarea en cuestión.

3. **Adaptación Contextual**: El modelo utiliza los ejemplos incrustados para entender los requisitos de la tarea y generar respuestas apropiadas. Los ejemplos few-shot ayudan al modelo a centrarse en aspectos específicos de la tarea, como el formato, el tono y el contenido.

4. **Refinamiento Iterativo:** Ajustar el prompt modificando los ejemplos y su presentación en función de las salidas del modelo. Este proceso iterativo ayuda a optimizar el prompt para un mejor rendimiento. Aplicación Práctica en Chatbots

Para implementar el prompting few-shot en chatbots, condensando lo expuesto por Jurafsky y Martin [2, p. 212], y, Bengio, Simard y Courville [6], se deben seguir los siguientes pasos:

1. **Identificación de Tareas Comunes**: Determinar las tareas que el chatbot manejará con frecuencia, como responder preguntas frecuentes, proporcionar recomendaciones o realizar transacciones.

2. **Selección de ejemplos**: Recopilar ejemplos de buena calidad para cada tarea. Estos ejemplos deben cubrir varios escenarios que el chatbot pueda encontrar.

3. **Diseño de Prompts**: Crear prompts incorporando los ejemplos para cada tarea. Asegurarse de que los prompts sean claros y concisos, proporcionando suficiente contexto para que el modelo entienda la tarea.

4. **Probar y Optimizar**: Probar el chatbot con los prompts diseñados y recopilar retroalimentación. Optimizar los prompts en función del desempeño del chatbot y la retroalimentación de los usuarios para mejorar su efectividad.

Al aprovechar el prompting few-shot, el aprendizaje en contexto y el ajuste de instrucciones, los chatbots pueden hacerse más versátiles y capaces de manejar una amplia gama de tareas con un entrenamiento adicional mínimo. Este enfoque no solo mejora el desempeño del chatbot, sino que también lo hace más adaptable a nuevos y cambiantes requisitos.

Programación de Prompts

El artículo "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm" presenta otros métodos de programación de prompts, desafiando la efectividad del paradigma few-shot y proponiendo enfoques alternativos como los zero-shot prompts y los metaprompts [11].

1. **Éxito de los Zero-shot Prompts**: Los zero-shot prompts, cuando se diseñan adecuadamente, pueden igualar o superar el rendimiento de los few-shot prompts. Esto destaca la importancia de la programación de prompts como una herramienta para guiar a los modelos de lenguaje.

2. **Contaminación Semántica en Few-shot Prompts**: Los ejemplos en few-shot prompts pueden introducir contaminación semántica, disminuyendo el rendimiento en algunas tareas. Es indispensable seleccionar y estructurar cuidadosamente los ejemplos para evitar este problema.

3. **Metaprompt Programming**: Los metaprompts permiten que el modelo genere prompts específicos para tareas complejas. Estos pueden ser útiles en preguntas cerradas que requieren un razonamiento secuencial y detallado. Los metaprompts actúan como una "semilla" que permite al modelo desarrollar un prompt más detallado y específico para la tarea específica.

Sistemas de Diálogo basados en Reglas y en Corpus

Chatbots Basados en Reglas

Estos sistemas responden a entradas de usuarios utilizando un conjunto predefinido de reglas. Aunque son fáciles de implementar y mantener, tienen limitaciones en términos de flexibilidad y escalabilidad, ya que solo pueden manejar situaciones específicas y predefinidas [2, p. 302].

Chatbots Basados en Corpus

Utilizan grandes conjuntos de datos para aprender a generar respuestas de manera más natural y contextualmente adecuada. Emplean modelos de aprendizaje automático para comprender y responder de manera más precisa, aunque requieren procesamiento y entrenamiento más complejos [2, p. 304].

Llenado de espacios (Slot Filling)

El llenado de espacios (Slots Fillings) es una técnica en sistemas de diálogo que implica la extracción de información específica de la entrada del usuario para completar espacios predeterminados en un marco de diálogo. Esta técnica asegura la coherencia y continuidad en la interacción, permitiendo que el sistema recopile toda la información necesaria de manera precisa. Por ejemplo, en el chatbot de cualificación de SOFIRI, los espacios podrían incluir el nombre del

solicitante, el programa de interés, las calificaciones académicas y las fechas límite de presentación [2, pp. 314-316].

El proceso de slot filling puede formalizarse utilizando modelos de aprendizaje supervisado, en los cuales cada palabra o secuencia de palabras en la entrada del usuario se clasifica en una categoría predefinida correspondiente a un slot. Una de las aproximaciones más comunes para esta tarea es el uso de modelos de etiquetado de secuencias, como los Modelos de Markov Ocultos (HMM) o las Redes Neuronales Recursivas (RNN), así como los avanzados Modelos de Lenguaje basados en transformers.

Política de Diálogo

La política de diálogo define las reglas y estrategias que el chatbot utiliza para gestionar la conversación. Esto incluye decidir cuándo pedir más información, cómo manejar respuestas ambiguas y cómo guiar al usuario de forma eficaz a través de la interacción, para conseguirlo debe agotar algunas instancias que le permitan cumplir con su propósito, por ejemplo, el chatbot puede necesitar hacer preguntas de seguimiento para recopilar información faltante o aclarar respuestas ambiguas. Las estrategias de solicitud ayudan a asegurar que los espacios necesarios se llenen con datos precisos [2, p. 316].

Cuando la entrada del usuario es poco clara o incompleta, el chatbot debe emplear estrategias para resolver las ambigüedades. Esto puede implicar hacer preguntas aclaratorias o hacer suposiciones razonables basadas en el contexto y debe al final **e**valuar la efectividad de un sistema de relleno de espacios, que implica medir su capacidad para recopilar la información requerida de manera precisa. Los indicadores clave de rendimiento incluyen la precisión del relleno de espacios, la satisfacción del usuario y la capacidad del sistema para manejar entradas complejas o inesperadas.

GUS: Sistemas de Diálogo Basados en Marcos Simples

El sistema GUS (General User Simulation) se utiliza para construir sistemas de diálogo que emplean una estructura de control basada en marcos (frames) para gestionar las interacciones. Estos sistemas son ideales para chatbots diseñados para tareas definidas, donde es necesario identificar y extraer información clave de las interacciones del usuario para completar formularios o procesos. La estructura de control en GUS se basa en la definición de marcos que representan los diferentes aspectos o componentes de la interacción. Cada marco contiene slots que deben ser llenados con información proporcionada por el usuario.

Determinación del Dominio e Intención

El sistema debe determinar en qué dominio se encuentra la interacción, es decir, el contexto general de la conversación.

El sistema debe identificar la intención del usuario, lo cual implica comprender qué está tratando de lograr el usuario con su consulta (por ejemplo, obtener información sobre los requisitos de admisión).

Evaluación de Sistemas de Lenguaje Natural

Evaluar la eficiencia del sistema implica medir su capacidad para completar tareas específicas de manera óptima. Esto incluye la precisión en el relleno de slots, la satisfacción del usuario y la tasa de éxito en la realización de tareas.

Basado en la evaluación el sistema puede ajustarse y mejorarse continuamente para optimizar su desempeño. Esto puede incluir ajustes en la política de diálogo, mejoras en los modelos de lenguaje utilizados para la comprensión y generación de respuestas, y actualizaciones en la base de datos de documentos para la recuperación de información.

Evaluación de Sistemas de Diálogo

Evaluar el desempeño de los chatbots y sistemas de diálogo es importante para asegurar su calidad y eficiencia. La evaluación implica medir la precisión, relevancia y coherencia de las respuestas generadas por el sistema.

Evaluación de Chatbots

¿BLEU o ROUGE?

Las métricas BLEU (Bilingual Evaluation Understudy) y ROUGE (Recall-Oriented Understudy for Gisting Evaluation) se utilizan para evaluar la calidad de las respuestas generadas por los chatbots comparándolas con respuestas de referencia. BLEU mide la precisión al comparar las secuencias de n-gramas en las respuestas generadas y las de referencia. ROUGE, por otro lado, mide la cobertura y la recuperación de las secuencias de n-gramas.

Evaluación de Diálogos Basados en Tareas

Evaluar diálogos basados en tareas implica medir la capacidad del sistema para completar tareas específicas de manera efectiva. Esto incluye evaluar la precisión de las respuestas, la satisfacción del usuario y la tasa de éxito en la ejecución de tales tareas.

Método de Levenshtein para Medir la Similitud entre Palabras

La distancia de Levenshtein es una métrica utilizada en el campo del NLP para medir la similitud entre dos cadenas de caracteres, comúnmente palabras. Esta métrica cuantifica el número mínimo de operaciones necesarias para transformar una cadena en otra, donde las operaciones permitidas incluyen inserciones, eliminaciones y sustituciones de caracteres [12].

Fundamentos del Cálculo

El cálculo de la distancia de Levenshtein se basa en una matriz de distancias que se construye para evaluar la similitud entre los prefijos de las dos cadenas comparadas. El procedimiento se

desarrolla mediante programación dinámica [13], que descompone el problema en subproblemas más simples y resuelve cada uno de ellos secuencialmente para alcanzar la solución global [14].

1. Inicialización de la Matriz: La matriz se inicializa con los índices de los caracteres de ambas palabras más uno, para incluir la comparación con el prefijo vacío.

2. Rellenado de la Matriz: Cada celda de la matriz representa la distancia entre los prefijos de las dos palabras. Esta se calcula tomando el mínimo de tres posibles valores:

    - La distancia entre los prefijos anteriores más uno (para inserciones).
    - La distancia entre los prefijos actuales más uno (para eliminaciones).
    - La distancia entre los prefijos anteriores más el costo de sustitución, que es cero si los caracteres coinciden y uno si no.

El valor final en la esquina inferior derecha de la matriz corresponde a la distancia de Levenshtein entre las dos palabras completas [15].

Ejemplo de Aplicación

Consideremos las palabras "hello" y "kelm". La matriz de distancias se llenaría reflejando un cambio de la k por la h, la m por la l y agregando la o, en total 3 cambios.

		h	e	l	l	o
	0	1	2	3	4	5
k	1					
e	2					
l	3					
m	4					

		h	e	l	l	o
	0	1	2	3	4	5
k	1	**1**	2	3	4	5
e	2	2	**1**	2	3	4
l	3	3	2	**1**	2	2
m	4	4	3	2	2	**3**

Basado en el ejemplo expuesto la distancia de Levenshtein entre "hello" y "kelm" es 3, lo que indica que se requieren tres operaciones de sustitución para convertir una palabra en la otra.

Aplicaciones en el Procesamiento de Lenguaje Natural

La distancia de Levenshtein se utiliza en varias aplicaciones dentro del PLN, tales como:

- Corrección ortográfica: Identificación y corrección de errores tipográficos en textos [16].
- Búsqueda aproximada: Encontrar palabras similares en grandes bases de datos textuales [17].

Embeddings de Word2Vec y FastText

Otro enfoque común para medir la similitud de textos generados es el uso de embeddings generados por modelos como Word2Vec y FastText. Estos modelos generan vectores densos para palabras, que pueden ser utilizados para medir similitudes de varias maneras:

- *Skip-Gram y Negative Sampling (SGNS)*: Word2Vec, por ejemplo, utiliza el modelo skip-gram con muestreo negativo para entrenar embeddings que maximizan la probabilidad de palabras de contexto reales mientras minimizan la probabilidad de palabras de ruido.

- *Similitud Coseno: la similitud entre textos puede ser medida usando la similitud coseno entre los vectores de embedding promedios de las palabras en cada texto.*

- *Sumas de n-gramas*: FastText mejora sobre Word2Vec al considerar sub-palabras (n-gramas), lo cual es particularmente útil para manejar palabras no vistas durante el entrenamiento.

Evaluación de Precisión

Para evaluar la precisión de los textos generados, se utilizan varias métricas y métodos cómo la Similitud Coseno: La similitud del coseno es una medida utilizada para calcular cuán similares son dos vectores en un espacio multidimensional, independientemente de su magnitud. Este concepto es muy útil en el procesamiento de lenguaje natural y en la minería de datos, entre otras áreas.

La similitud del coseno [2, p. 112] entre dos vectores AA y BB se define como el coseno del ángulo entre ellos. Matemáticamente, se expresa como:

$$Similitud\ (A,B) = \cos\theta = \frac{A \cdot B}{|A||B|}$$

donde:

A·B es el producto punto de los vectores A y B.

‖A‖ y ‖B‖ son las normas (o longitudes) de los vectores A y B respectivamente.

θ es el ángulo entre los vectores.

El producto punto A·B se calcula como:

$$A \cdot B = \sum_{i=1}^{n} A_i B_i$$

donde A_i y B_i son los componentes de los vectores A y B.

La norma de un vector A se calcula como:

$$\|A\| = \sqrt{\sum_{i=1}^{n} A_i^2}$$

Interpretación

Valor de 1: Indica que los vectores son idénticos.
Valor de 0: Indica que los vectores son ortogonales (no tienen similitud en ninguna dirección).
Valor de -1: Indica que los vectores son diametralmente opuestos.

BERTScore

Uno de los métodos más destacados para medir la precisión en la generación de texto es BERTScore. Este método utiliza el modelo BERT (Bidirectional Encoder Representations from Transformers) para calcular embeddings contextuales en una oración.

Los pasos para calcular BERTScore son los siguientes:

1. Tokenización y Embeddings: Se tokenizan tanto la oración de referencia como la generada por el modelo y cada token se convierte en un vector de embedding usando BERT.

2. Similitud Coseno: Se calcula la similitud coseno entre cada par de tokens (uno de la referencia y otro de la oración generada). Esta métrica mide cuán alineados están los vectores en un espacio multidimensional.

3. Matching: Se realiza un emparejamiento avaro (greedy matching) para alinear cada token en la referencia con el token más similar en la oración generada, y viceversa, para calcular el recall y la precisión.

4. Cálculo de F1: Finalmente, se combina la precisión y el recall en una medida F1, que proporciona un balance entre ambas métricas.

$$F1_{score} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

CAPÍTULO 5. ESTADO DEL ARTE

Modelos de Lenguaje y Transformers

Los LLMs utilizan la arquitectura transformer, destacándose por su capacidad de manejar grandes volúmenes de datos y aprendizaje a partir de estos. Estos modelos han demostrado un rendimiento sin precedentes en una variedad de tareas de NLP, desde la clasificación hasta la traducción de texto. La personalización mediante técnicas de fine-tuning es un factor importante para adaptar estos modelos generales a necesidades específicas, mejorando significativamente la interacción y funcionalidad de los chatbots.

Despliegue y Optimización de LLMs

La implementación de LLMs en la nube ha permitido escalar y mejorar la accesibilidad de esta tecnología, facilitando su integración en sistemas de chatbots que requieren capacidad de respuesta en tiempo real y acceso continuo. La mejora continua de los modelos a través de ajustes a la medida y el uso de APIs específicas para LLMs son prácticas comunes que ayudan a mantener la coherencia y desempeño de los sistemas basados en estos modelos.

Desafíos y Oportunidades Futuras

A pesar del reciente avance en esta disciplina, la implementación de NLP en chatbots enfrenta desafíos como la adaptabilidad a contextos específicos y la gestión de la diversidad lingüística. Investigaciones futuras en modelos más robustos y técnicas avanzadas como el aprendizaje en contexto y la personalización de modelos son indispensables para superar estas limitaciones.

Contribuciones a la Investigación y Aplicaciones Prácticas

La contribución esencial de este proyecto radica en la metodología adoptada para la definición de los objetivos del chatbot antes de proceder a su desarrollo. Al delimitar el propósito del chatbot desde las etapas iniciales, se establecen las bases para seleccionar las tecnologías, modelos y arquitecturas adecuadas para satisfacer los requerimientos particulares. Este enfoque dirigido asegura que el chatbot desarrollado no solo cumpla con las funciones básicas esperadas, sino que también responda de a los requisitos particulares del contexto para el que fue creado.

CAPÍTULO 6. METODOLOGÍA

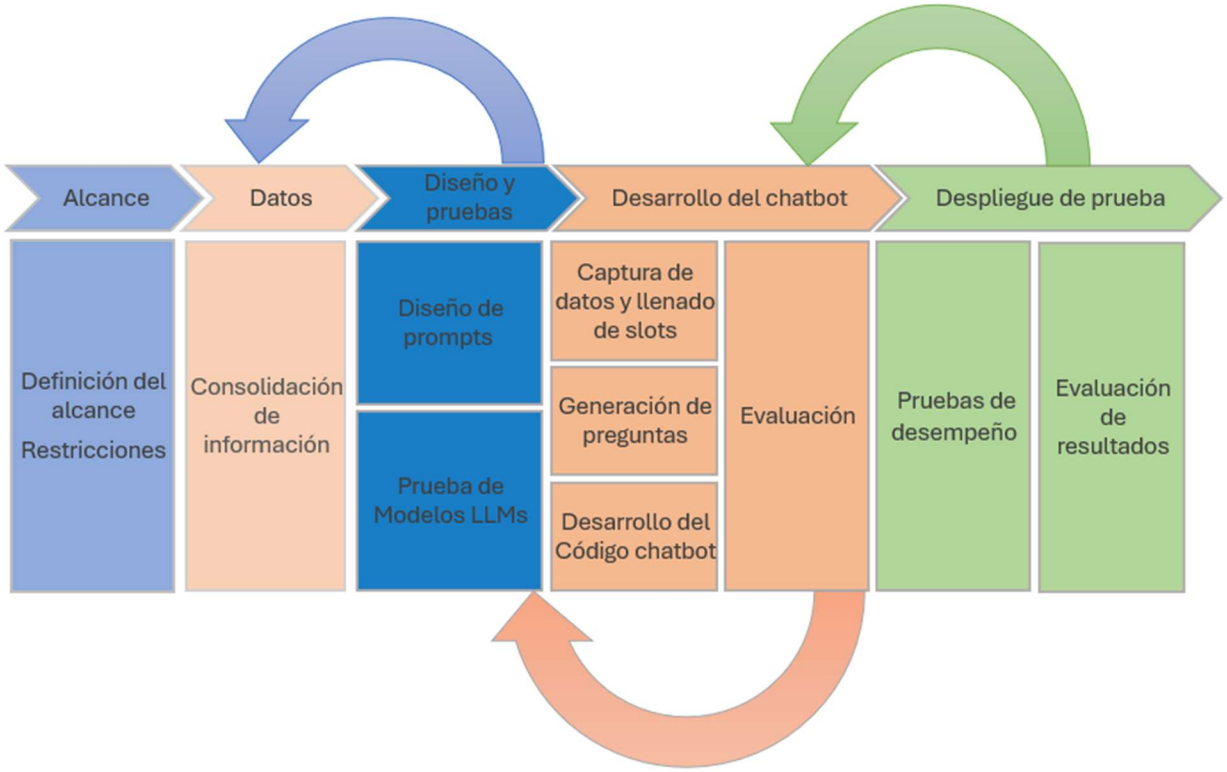

Figura 6. Metodología del ciclo del proyecto, adaptado de GenerativeAI Project Lifecycle

La metodología propuesta para el desarrollo de este trabajo se basa en una serie de pasos que aseguran el desarrollo y la implementación del chatbot en un entorno de prueba. Estos pasos abarcan desde el estudio inicial de algoritmos y técnicas recientes en el NLP hasta la validación de resultados. A continuación, se describen las herramientas y los datos necesarios para llevar a cabo cada una de estas etapas.

Definición y delimitación del alcance

Límites específicos del alcance del chatbot

Las interacciones con el chatbot deben resultar en actividades específicas y en cada una de ellas se limitará a:

- **Recopilación de datos de usuario:** Especificar la información exacta que los usuarios deben proporcionar al chatbot durante las interacciones.
- **Llenado de slots**: Implementar un proceso para capturar y almacenar los datos de los usuarios en slots durante la interacción, generando preguntas coherentes para asegurar que toda la información necesaria sea obtenida.
- **Evaluación del desempeño**: Evaluar el desempeño del chatbot en la captura de datos para el llenado de slots, utilizando métricas que midan la similitud entre vectores de embeddings.
- **Evaluación de generación de preguntas**: Validar la calidad y coherencia de las preguntas generadas durante las interacciones con los usuarios.
- **Limitación de interacciones:** Restringir las interacciones del chatbot a la generación de preguntas relacionadas específicamente con los slots que se busca llenar. El chatbot no debe generar información adicional ni responder preguntas para evitar comprometer a la compañía con información proporcionada.
- **Idioma de las interacciones**: Aunque los modelos utilizados pueden manejar múltiples idiomas, tanto las preguntas como las interacciones realizadas por el chatbot se harán en inglés.

Estudio de Algoritmos y Técnicas NLP

A continuación, se debe adelantar una fase de investigación y estudio en la que se exploren algoritmos, técnicas y avances recientes en el campo del NLP. Además, se debe investigar modelos preentrenados basados en transformers, como Google/T5, OpenAI/GPT-3.5, OpenAI/GPT-4o y Meta/Llama2, para determinar cuál es la opción más adecuada para el contexto del proyecto.

Obtención y Estructuración de Información

La información disponible fue determinante en el desarrollo del chatbot de SOFIRI. Por cumplimiento de sus políticas corporativas, la compañía había recopilado suficiente información, indispensable para la evaluación de los prompts y los modelos de lenguaje (LLMs), así como para validar la coherencia y similitud de los datos capturados. Desde 2018, SOFIRI ha acumulado más

de 500,000 registros de usuarios a través de diversas plataformas, incluyendo Facebook Messenger, aplicaciones móviles Android e iOS, y el sitio web oficial de SOFIRI PTY LTD.

Fuentes de datos

Los datos sobre los cuales se prepararon los tokens, los slots y datos de prueba se recopilaron desde el bot existente en SOFIRI PTY LTD desde los siguientes orígenes:

- Facebook Messenger: Interacciones realizadas a través del chatbot de SOFIRI en Facebook Messenger https://www.facebook.com/sofiri/.
- Aplicaciones Android e iOS: Interacciones con el chatbot en las aplicaciones móviles de SOFIRI https://play.google.com/store/apps/details?id=com.sofiri.app y https://apps.apple.com/app/id1514039250.
- Aplicación web de SOFIRI PTY LTD: Registros de conversaciones entre usuarios y el chatbot en el sitio web oficial de la compañía https://sofiri.com/student/.

Confidencialidad de los datos

La información recopilada y utilizada para este proyecto es estrictamente confidencial. Todos los datos de las interacciones de los usuarios con el chatbot, tanto en Facebook Messenger como en las aplicaciones móviles y la web de SOFIRI, están protegidos y se manejan con altos estándares de seguridad para garantizar la privacidad y protección de la información personal de los usuarios.

Estructuración de datos

Para facilitar el procesamiento y análisis, es necesario consolidar archivos en formato .csv que contengan las preguntas y respuestas específicas requeridas para el llenado de slots en el proceso de cualificación de aspirantes. Estos archivos se deben organizar de la siguiente manera:

- Archivos por slot: Cada archivo .csv corresponde a un slot específico y contiene todas las respuestas proporcionadas por los usuarios desde 2018.

- Campos estructurados: Cada archivo incluye campos como Fecha de respuesta y Respuesta.

Preprocesamiento de los Conjuntos de Datos

Una vez recopilada la información, se procede con la limpieza y preprocesamiento de los datos para asegurar su calidad y relevancia antes de utilizarlos en el entrenamiento de los modelos de lenguaje. Este proceso incluye varias etapas:

- Limpieza de Datos:
  - Eliminación de registros incompletos.
  - Corrección de errores tipográficos y gramaticales.
- Normalización:
  - Estandarización de formatos.
  - Conversión de texto a minúsculas.
- Filtrado y Remoción de Ruido:
  - Filtrado de contenido irrelevante.
  - Segmentación de oraciones.

Evaluación de Prompts y Modelos LLM

Durante esta etapa, se deben diseñar los prompts para cada una de las preguntas “slots” y es necesario probarlos con los modelos de lenguaje (LLMs) utilizados en el chatbot. Esta evaluación incluye la validación de la coherencia y similitud de los datos capturados y la calidad de las respuestas generadas por el modelo.

Durante la fase de desarrollo del chatbot de SOFIRI, una etapa crítica es la evaluación de prompts y modelos de lenguaje (LLMs). Esta fase es importante para asegurar que el chatbot pueda interactuar con los usuarios y captar la información necesaria.

Para la evaluación de modelos de lenguaje para el llenado de slots “slot filling” del chatbot, se diseñan una serie de prompts. Estos prompts siguen principios clave que buscan garantizar la precisión. A continuación, se describen los principios generales utilizados en la construcción de estos prompts:

- **Claridad y Precisión**: Los prompts están diseñados para ser claros y específicos, proporcionando instrucciones detalladas y ejemplos concretos para asegurar que el modelo entienda correctamente la tarea a realizar.
- **Consistencia**: Se mantiene una estructura uniforme en todos los prompts, utilizando roles definidos (sistema, usuario, asistente) para guiar la interacción y facilitar la comprensión del modelo.
- **Generalización**: Los prompts se construyen para ser aplicables a una variedad de entradas de usuario, asegurando que el modelo pueda manejar diferentes formas de expresar la misma información.

Los prompts son entradas específicas diseñadas para guiar al modelo de lenguaje a generar respuestas oportunas. Estos prompts son diseñados para cubrir las preguntas que el chatbot debe hacer a los usuarios para recopilar información deseada. A continuación, se presenta un ejemplo de un prompt:

Figura 7. Diseño de prompt de inicio "saludo usuario"

El diseño de este tipo de prompts se fundamenta en la evaluación recurrente de muestras. En primer lugar, los modelos de lenguaje como GPT-3.5 y GPT-4o se entrenan utilizando grandes corpus de texto, lo que les permite aprender patrones y estructuras del lenguaje humano [18]. Los prompts sirven como guías específicas que orientan al modelo a focalizarse en ciertos tipos de entradas y generar respuestas contextualizadas y coherentes [2, p. 311].

Evaluación de Modelos de Lenguaje

Una vez construidos los prompts, se procede a evaluar varios modelos de lenguaje preentrenados, como GPT-3.5, GPT-4o y Llama2. Estos modelos son seleccionados debido a su capacidad para manejar tareas de procesamiento de lenguaje natural. La evaluación se centra en los siguientes aspectos:

- Coherencia de la generación de preguntas: Se verifica que las preguntas generadas por los modelos sean coherentes para las preguntas formuladas en los prompts.
- Precisión en la Captura de Datos: Se mide la precisión con la que los modelos pueden extraer la información requerida (por ejemplo, saludos, ciudades, niveles de estudio) a partir de las respuestas de los usuarios.

- Similitud entre Vectores de Embeddings: Se utilizan métricas para evaluar la similitud entre los vectores de embeddings generados por las preguntas del modelo y las preguntas esperadas, asegurando que las preguntas generadas sean contextualmente adecuadas.

Despliegue de Prueba

Se realizará un despliegue piloto del chatbot en una instancia EC2 (Elastic Compute Cloud) de AWS (Amazon Web Services) en un subdominio corporativo, para pruebas con un grupo razonable de participantes, de al menos 30. Este número se considera prudente debido a que se trata de una prueba piloto, y posteriormente se implementará una versión de prueba por parte de SOFIRI. Esta fase permitirá evaluar el desempeño del chatbot en un entorno real y recopilar retroalimentación para realizar ajustes y mejoras.

Validación de Resultados

Se emplearán técnicas de validación cruzada sobre las preguntas, para medir la calidad de las preguntas generadas por el chatbot y la comparación con los tokens recuperados de las respuestas de los evaluados para validar los slots del chatbot. Los datos de validación incluirán respuestas proporcionadas por usuarios en un simulacro del proceso de cualificación, comparándose con las respuestas del chatbot.

Desarrollo del Chatbot

El diagrama de proceso adjunto describe el flujo de trabajo de un Chatbot basado en sistemas de diálogo con llenado de slots utilizando Modelos de Lenguaje Grandes (LLMs) y técnicas avanzadas de NLP. A continuación, se presenta una descripción detallada de cada paso del diagrama, basada en los conceptos de NLP y técnicas de slot filling discutidas previamente.

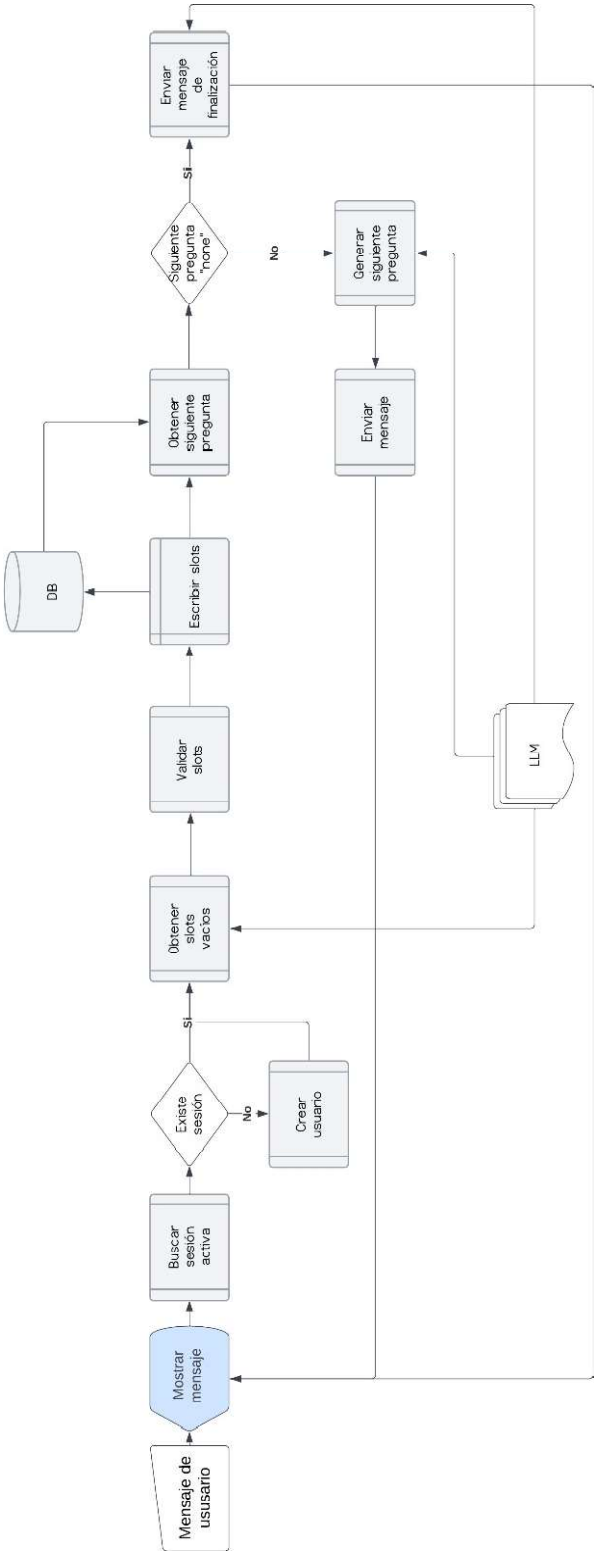

Figura 8. Diagrama de flujo del Proceso del Chatbot

A continuación, se explican cada una de las funciones que componen el Chatbot

Mensaje del usuario

El proceso comienza con la entrada del usuario. Esta entrada puede ser una pregunta, una declaración o cualquier tipo de interacción textual. En el contexto de un Chatbot de ingreso a universidades internacionales, el usuario podría preguntar sobre los requisitos de admisión, programas académicos disponibles, fechas importantes, entre otros.

Buscar sesión activa

El sistema busca si el identificador de sesión está asociado a un registro de la tabla de usuarios en caso de estarlo continúa, en caso opuesto crea un nuevo usuario en la base de datos.

Obtener Slots vacíos

El sistema identifica los slots (campos de información) que aún no han sido completados. Los slots son entidades específicas que el chatbot necesita rellenar para proporcionar una respuesta completa y precisa. Ejemplos de slots pueden incluir el nombre del usuario, el programa de interés, la fecha de inicio, entre otros. Esta técnica es comúnmente utilizada en sistemas de diálogo basados en frames como GUS (Generalized User Simulation)

Validar Slots

El Chatbot valida los slots que ya han sido llenados para asegurarse de que la información capturada es correcta y coherente. Esta validación puede incluir verificaciones de formato, coherencia y pertinencia de los datos proporcionados. Técnicas avanzadas como los Transformers se utilizan aquí para mejorar la precisión del reconocimiento y validación de entidades

Escribir Slots

Una vez se obtienen los valores válidos para los slots que se puedan llenar a partir del mensaje suministrado por el usuario, son registrados en la base de datos junto con el número de tokens que fueron necesarios para obtenerlos.

Obtener siguiente pregunta

El sistema genera la siguiente pregunta basada en los slots aún vacíos y la información capturada hasta el momento. Este paso es crucial para guiar al usuario a proporcionar toda la información necesaria de manera estructurada. Los LLMs son utilizados para generar preguntas relevantes y contextualmente adecuadas.

El chatbot evalúa si no hay más preguntas que hacer. Si todos los slots necesarios han sido llenados, el proceso avanza hacia el envío de un mensaje final. Si aún quedan slots por llenar, se genera la siguiente pregunta.

- Yes (Sí): Si no hay más preguntas que hacer, el sistema procede a enviar un mensaje final al usuario.
- No (No): Si todavía hay slots vacíos, el sistema genera la siguiente pregunta para obtener la información restante.

Generar siguiente pregunta.

El chatbot envía la siguiente pregunta al usuario, continuando la interacción hasta que todos los slots relevantes hayan sido llenados. Los LLMs se utilizan para generar preguntas claras y contextualmente adecuadas que faciliten la obtención de la información necesaria

Enviar mensaje de finalización.

Una vez que todos los slots necesarios han sido completados y validados, el chatbot envía un mensaje final al usuario. Este mensaje puede incluir un resumen de la información capturada, próximos pasos, o cualquier otra información relevante.

CAPÍTULO 7. RESULTADOS Y DISCUSIÓN

A continuación, se presenta el "Formulario de Slots para Captura de Datos de Usuarios", diseñado para recopilar información de los usuarios del chatbot. Este formulario incluye preguntas que abarcan desde datos personales básicos hasta detalles específicos sobre el curso y las expectativas de pago. Cada pregunta está identificada por un Slot_ID único que facilita el procesamiento y organización de las respuestas. Las preguntas han sido proporcionadas por SOFIRI.

Tabla 2. Formulario de Slots para Captura de Datos de Usuarios

Slot_ID	Pregunta base	Nombre de Slot
1	What is your given name?	**first_name**
2	What is your surname?	**last_name**
3	What city do you live in?	**location**
5	What degree or course did you study?	**user_degree**
7	What level would you like to study?	**intent_level**
8	What is the name of the course you would like to study?	**intent_course**
9	When do you want to begin studying?	**commencement**
10	How much do you expect to pay in course fees per year?	**budget**
11	Can you tell me your date of birth? (DD/MM/YYYY)	**date_of_birth**
12	And your country of citizenship?	**country_citizenship**
14	What's an email address I can send them to?	**email**

Resultados de la etapa de evaluación de prompts y modelos

En esta parte se presentan las conclusiones derivadas de la aplicación del chatbot a la población de prueba, el análisis y evaluación del llenado de slots (slot filling) utilizando diferentes modelos de lenguaje (LLMs).

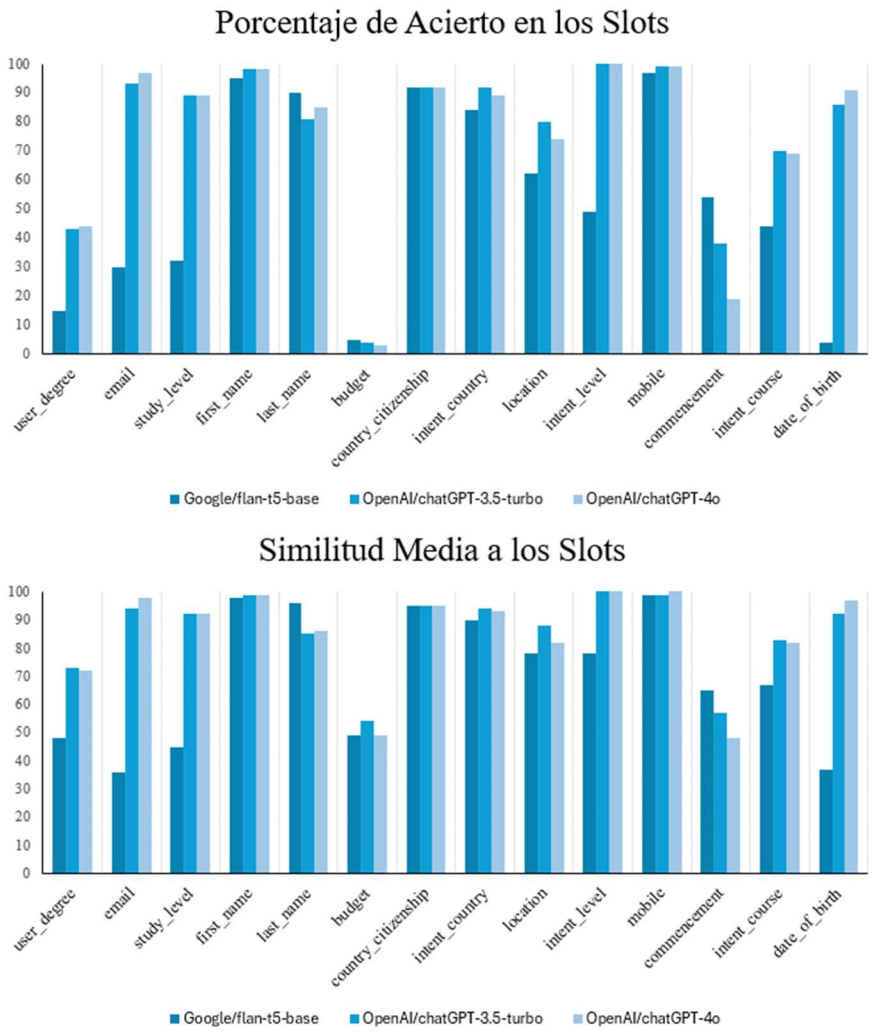

Figura 9. Gráfica de similitud y precisión

- Eficiencia del Modelo GPT-4o: Entre los modelos evaluados (google/flan-T5 small, chatGPT-3.5-turbo, y chatGPT-4o), el modelo chatGPT-4o demostró ser el más eficiente en términos de precisión y similitud de respuestas. Logró altos porcentajes de respuestas correctas y similitud en la mayoría de los slots, lo que indica su capacidad superior para interpretar y procesar los ingresos de los usuarios. Además, GPT-4o se destacó por su velocidad de procesamiento, lo que redujo los tiempos de espera en comparación con otros modelos.

- Desafíos y Mejoras en el Llenado de Slots: Se identificaron ciertas preguntas que presentaron desafíos para todos los modelos evaluados, como "What degree or course did

you study?" y "How much do you expect to pay in course fees per year?". Estos slots mostraron bajos porcentajes de respuestas correctas y similitud, lo que indica la necesidad de mejorar los prompts o considerar modelos adicionales para capturar esta información de manera más precisa.

- La confiabilidad de las APIs utilizadas fue un aspecto evaluado durante el proyecto. Específicamente, la API de Llama 2.0 y 3.0, proporcionada por Replicate [19], no mantuvo una ejecución estable. Replicate es una empresa que facilita el despliegue de modelos de aprendizaje automático a gran escala, ofreciendo APIs para diversos modelos de inteligencia artificial.

Resultados de la prueba piloto del chatbot

Resultados de la Pruebas de Desempeño

El promedio de la evaluación con la métrica Levenshtein normalizada para la variable Correct se elevó a 98.14% y para Similarity alcanzó 98.79% evidenciando que el chatbot propuesto en efecto consigue extraer información más precisa, en un menor tiempo apoyados en chatGPT-4º.

Del total de los slots, solo el Slot_9 (budget) reflejó un desempeño ligeramente inferior al promedio en cada métrica con un 87% de precisión y un 96% de similitud, lo que sugiere que, aunque el chatbot mejoró sustancialmente en la evaluación, existen algunas áreas de mejora en la captura de datos específicos relacionados con aspectos financieros.

Los resultados evidencian que el chatbot es altamente efectivo en la captura de datos correctos con una alta similitud en la mayoría de los slots, reforzando la viabilidad para su implementación en entornos reales en donde se requiere la captura precisa de información del usuario.

Resultados de la validación del modelo para la generación de preguntas

La siguiente gráfica muestra la evaluación de similitud de generación de preguntas del chatbot utilizando el modelo SBERT [20] distiluse-base-multilingual-cased-v2. En esta evaluación, se

comparan las preguntas generadas por el chatbot con las preguntas esperadas para capturar la información requerida de los usuarios. La similitud se mide en dos categorías: similitud con buenas preguntas y similitud con malas preguntas.

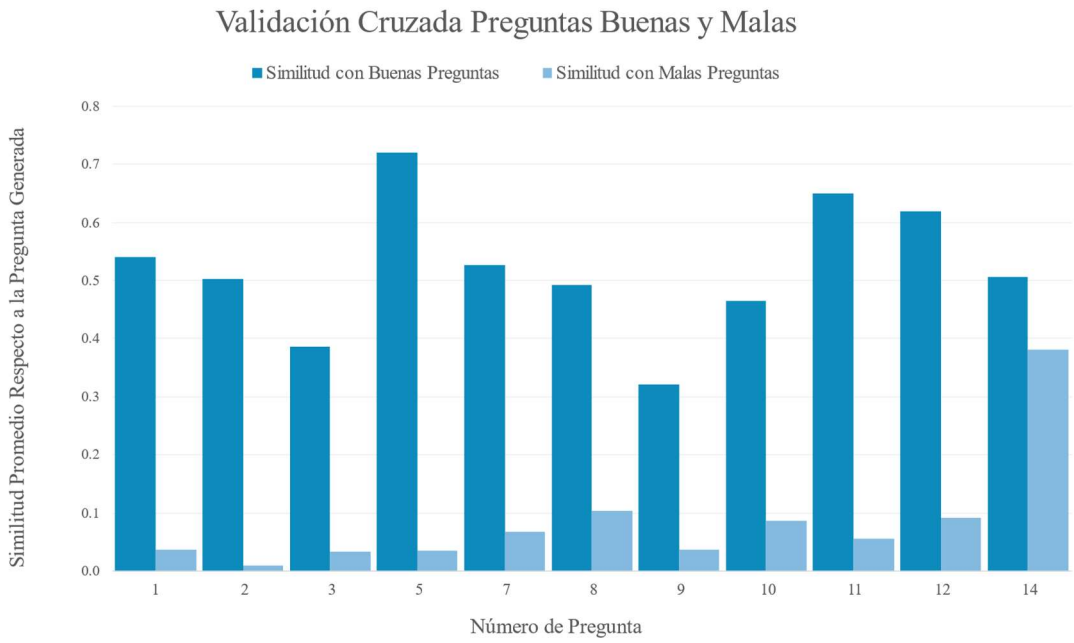

Figura 10.Validación cruzada de las preguntas generadas por el chatbot

Las preguntas 1, 2, 5, 7, 8, 11, 12, y 14 muestran una alta similitud en las buenas preguntas, con valores que oscilan entre 0.5 y 0.7. Esto indica que el chatbot es capaz de generar preguntas que son muy similares a las preguntas esperadas, lo que es crucial para capturar correctamente la información requerida de los usuarios.

Al finalizar la etapa de recuperación de muestras, cada interacción del chatbot requería evaluar 14 consultas en serie, resultando en retrasos de 40 a 60 segundos por contacto, afectando negativamente la experiencia del usuario. Para mitigar esto, se utilizó la librería Concurrent, para gestionar solicitudes asincrónicas.

Resultados del Análisis de costos

Costos del uso de Modelos en el chatbot.

Además de evaluar la precisión y la efectividad de los modelos de lenguaje (LLMs) para el proceso de slot filling, es importante considerar los costos asociados con los requerimientos de máquina necesarios para ejecutar cada modelo. Estos costos incluyen la cantidad de recursos computacionales necesarios, como memoria, capacidad de procesamiento, y tiempo de ejecución. A continuación, se presenta una comparación detallada de los costos de los requerimientos de máquina entre los modelos probados, y también el costo de su implementación en el chatbot:

Tabla 3. Comparación de costo uso de Modelos

Modelo	Costo x 1M Tokens de Entrada (USD)	Costo x 1M Tokens de Salida (USD)	Costos de Infraestructura (Estimados)	Observaciones
T5 Large	N/A	N/A	$3.06	Basado en instancia p3.2xlarge en AWS
T5 Extra Large	N/A	N/A	$12.24	Basado en instancia p3.8xlarge en AWS
Llama 2	$0.65	$2.75	$1.15 (basado en 1000 segundos)	Modelo de código abierto, costos de infraestructura
ChatGPT-3.5 Turbo	$0.50	$1.50	N/A	Costos basados en la API de OpenAI
ChatGPT-4º	$5.00	$15.00	N/A	Costos basados en la API de OpenAI

Al analizar los costos de implementación para procesar 1M de tokens entre los diferentes modelos, se observa que:

- ChatGPT-3.5 Turbo es el modelo más económico con un costo total de $2.00 por 1M tokens.

- ChatGPT-4o es el modelo más caro con un costo total de $20.00 por 1M tokens, lo cual es justificado por sus capacidades avanzadas y alta precisión, balanceando entre la cantidad y calidad de información que puede capturar y el tiempo al aire de exposición.

Evaluación de Costos por pregunta con ChatGPT-4o

A continuación, se enseñan los resultados del cálculo del costo asociado a la implementación de diferentes preguntas de slots utilizando el modelo ChatGPT-4o. En la siguiente tabla se presenta la descripción de los resultados obtenidos, incluyendo la cantidad de tokens utilizada y el costo por pregunta.

Tabla 4. Costo y cantidad de tokens por pregunta

Slot_ID	Descripción de la pregunta	Qty_tokens	Cost_question
1	What is your given name?	121	0.002
2	What is your surname?	123	0.002
3	What city do you live in?	123	0.002
5	What degree or course did you study?	126	0.002
7	What level would you like to study?	125	0.002
8	What is the name of the course you would like to study?	136	0.002
9	When do you want to begin studying?	178	0.003
10	How much do you expect to pay in course fees per year?	310	0.005
11	Can you tell me your date of birth? (DD/MM/YYYY)	162	0.002
12	And your country of citizenship?	135	0.002
14	What's an email address I can send them to?	205	0.003
Total		**1744**	**USD 0.0262**

Los resultados obtenidos muestran que el costo total para procesar las preguntas específicas del chatbot utilizando ChatGPT-4o es de USD 0.0262.

Para interactuar con un promedio anual de 50,000 usuarios, que representan la cantidad de episodios o usuarios que interactuaron a través de los canales de utilizando ChatGPT-4o, se estima que se necesitarán aproximadamente **87,200,000** tokens, con un costo total anual de **USD** $**1,310**. Este cálculo se basa en un uso promedio de 1744 tokens por usuario, con un costo de USD $0.0262 por interacción.

Cantidad promedio de intentos por Slot.

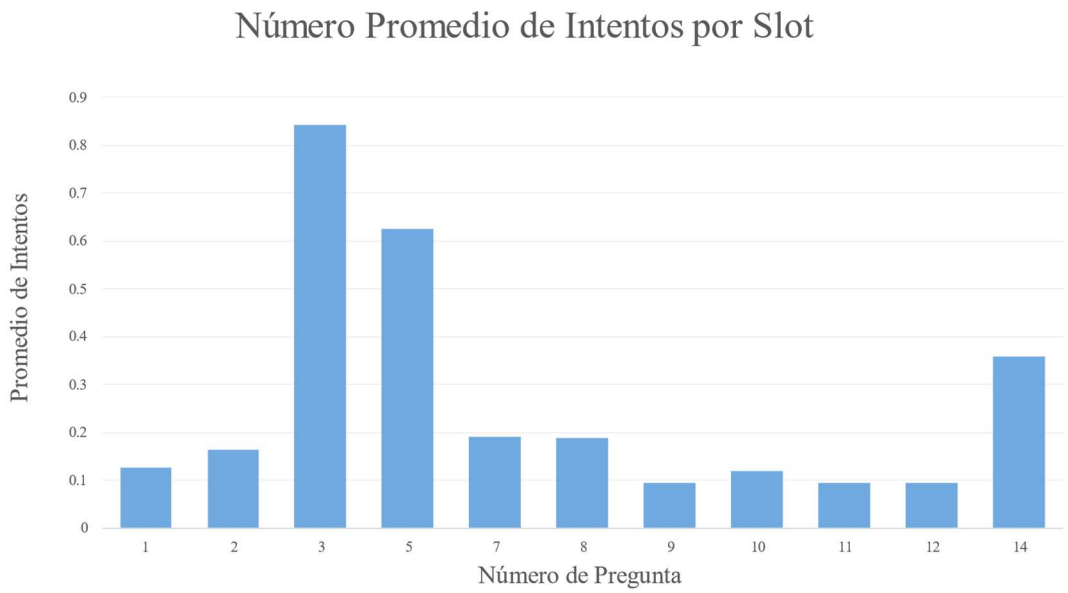

Figura 11. Cantidad promedio de intentos por slot

La gráfica anterior muestra el número promedio de intentos que los participantes requirieron durante la prueba para responder a una pregunta hasta que el chatbot logró llenar los slots con los datos capturados durante las interacciones.

Slot 14: Este slot, correspondiente a la pregunta "What's an email address I can send them to?", tiene el promedio más alto de intentos, cercano a 3. Esto indica que el chatbot tuvo más dificultades para obtener una dirección de correo electrónico válida, lo que sugiere que la pregunta podría haber sido ambigua o que los usuarios fueron cuidadosos al proporcionar esta información sensible.

Análisis de Preguntas Alcanzadas en la prueba del chatbot

La siguiente figura muestra el número de usuarios por última pregunta alcanzada durante las pruebas con 64 participantes y a continuación se presenta una descripción detallada de los resultados, dando cuenta que cerca del 72% de los participantes finalizó con éxito la prueba y el porcentaje restante se distribuyó como se muestra en la figura posterior, ahora bien, los slots se

presentan en orden ascendente de acuerdo con su numeración, dando a entender que el número de desertores no es muy alto en comparación con quienes lo terminaron, si llama la atención que lo hicieran en las preguntas iniciales.

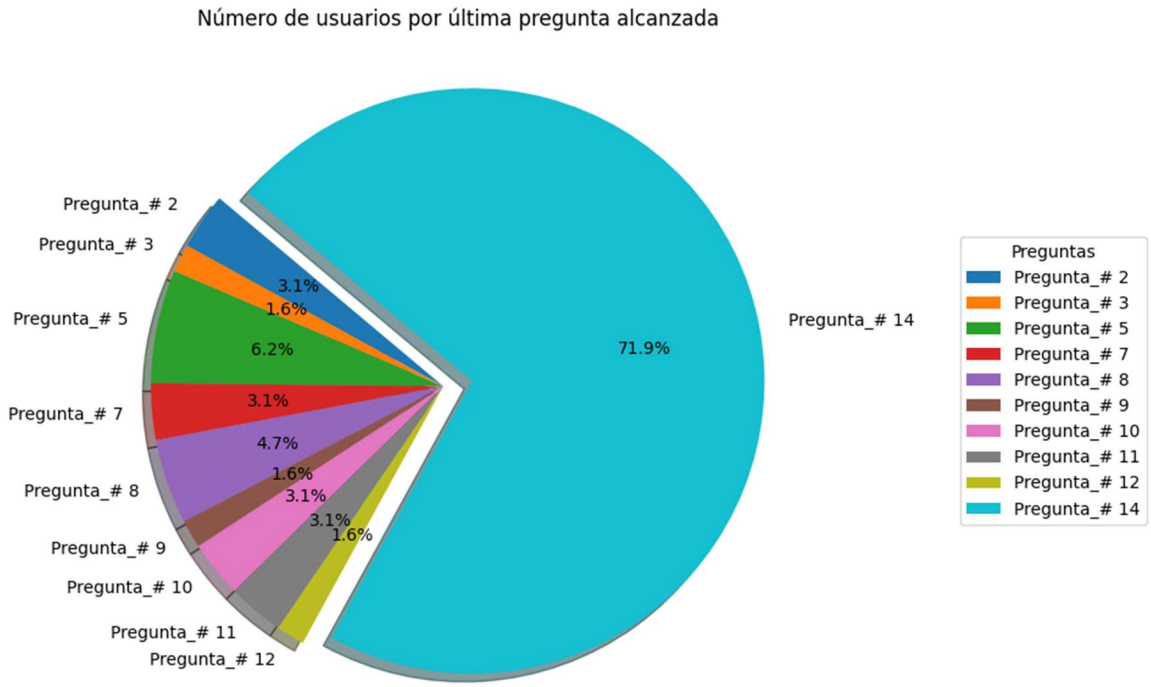

Figura 12. Usuarios por última pregunta alcanzada

- Eficiencia del Flujo de Preguntas: La mayoría de los participantes (71.9%) completaron el cuestionario, llegando hasta la pregunta final (#14: "What's an email address?"). Esto sugiere que el flujo de preguntas fue claro y efectivo, manteniendo a los usuarios comprometidos hasta el final de la interacción, lo cual es positivo considerando la diversidad de la población encuestada.

- Identificación de Preguntas complejas: Las preguntas #5 ("What degree or course did you study?") y #8 ("What is the name of the course you would like to study?") tuvieron tasas altas de abandono (6.2% y 4.7% respectivamente). Esto indica que estas preguntas pueden ser confusas o difíciles para los hispanohablantes, dado que el chatbot está diseñado para interactuar en inglés. En el futuro, si se requiere incorporar otros idiomas, será fundamental

revisar y adaptar la formulación de las preguntas. Esto incluye proporcionar ejemplos específicos, así como utilizar nombres de cursos y niveles de estudio en los idiomas pertinentes para mejorar la comprensión.

Resultados de la encuesta aplicada a los participantes de la prueba

A partir de la prueba del chatbot con un conjunto de 64 participantes, entre los cuales se incluyó a estudiantes de pregrado, estudiantes de postgrado y profesionales con cargos de dirección, se pueden extraer las siguientes conclusiones:

Figura 13. Resultados de la primera pregunta de la encuesta

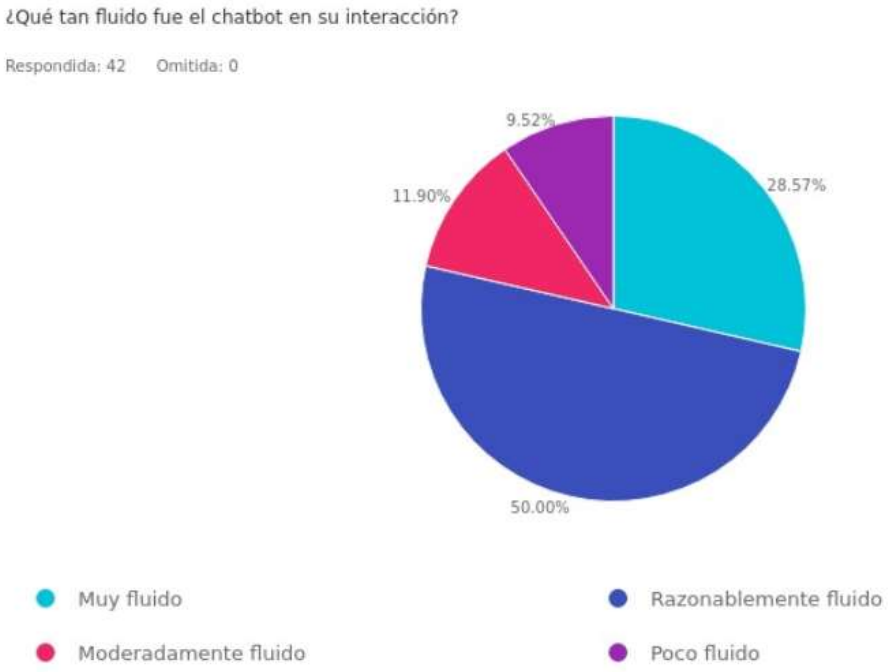

Figura 14. Resultados de la segunda pregunta de la encuesta.

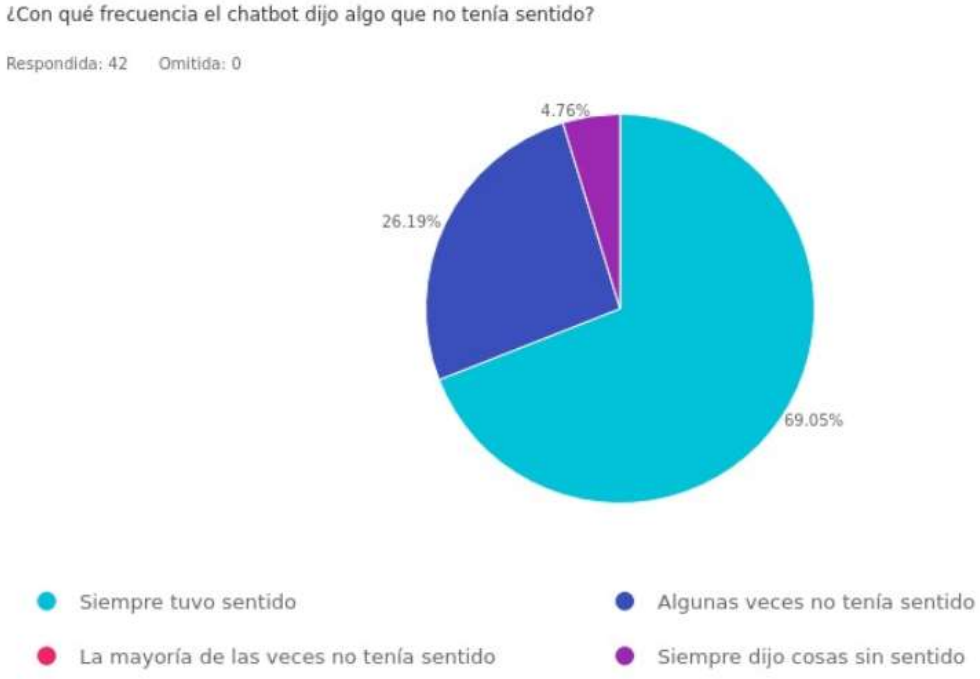

Figura 15. Resultados de la tercera pregunta de la encuesta.

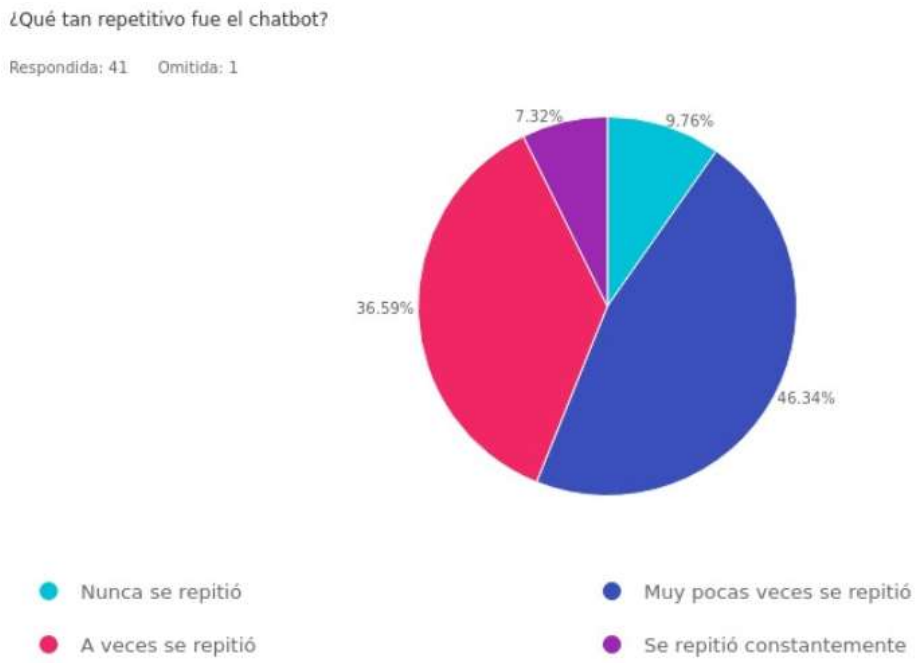

Figura 16. Resultados de la cuarta pregunta de la encuesta.

Con base en los resultados de las encuestas realizadas sobre la experiencia de uso del chatbot, se pueden extraer las siguientes observaciones:

- Facilidad y Fluidez de Uso: El chatbot ha demostrado ser efectivo en términos de facilidad y fluidez de interacción. Más del 57% de los usuarios encontraron la interacción con el chatbot más fácil en comparación con otros chatbots, y el 78% la consideraron razonablemente fluida o muy fluida. Esto sugiere que el chatbot está bien optimizado para facilitar la comunicación, ofreciendo una experiencia de usuario positiva.

- Coherencia y Variedad de Respuestas: La mayoría de los usuarios (69%) afirmaron que las respuestas del chatbot siempre tenían sentido, y el 56% indicó que el chatbot rara vez se repetía. Sin embargo, hay un pequeño margen de mejora ya que un 5% encontró que el chatbot siempre dijo cosas sin sentido, y un 7% señaló que se repetía constantemente. Estos resultados reflejan un alto grado de satisfacción con la coherencia y la diversidad de las respuestas, aunque se deben abordar las áreas mencionadas para mejorar aún más la experiencia del usuario.

CAPÍTULO 8. CONCLUSIONES

El desarrollo e implementación del chatbot basado en técnicas de NLP y LLM chatGPT-4o ha demostrado una mejora importante en la precisión de la captura de datos de los participantes en la prueba piloto. La comparación con los datos históricos mostró que el nuevo chatbot es capaz de completar los "slots" con una mayor exactitud y con un número menor de intentos.

Los resultados de la encuesta aplicada a la población de muestra, arrojó que el chatbot ofreció una interacción más fluida y los resultados de las métricas concluyen que también es más eficiente y se espera una reducción a la tasa de abandono observada en el sistema anterior.

Después de la implementación del nuevo chatbot, las métricas arrojaron que las preguntas generadas son similares a las formuladas por un experto, además, a medida en que se avanza en la sesión con el usuario, el chatbot adquiere el contexto necesario para interactuar de manera más natural, brindándole una mejor experiencia de usuario al participante por la personalización de las preguntas que genera.

Dado que la arquitectura del nuevo chatbot se desarrolló bajo las especificaciones de SOFIRI, este chatbot es naturalmente integrable con el software de la compañía y se probó apoyado en la nube de la empresa y los resultados aquí presentados son un producto extraído de forma directa del entorno corporativo en el ambiente de pruebas.

Retos

Ajuste fino para mejorar la precisión en las preguntas en las que el llenado de slots tuvo un bajo desempeño y para que identifique la geolocalización del usuario con el que interactúa y personalice la experiencia según las particularidades regionales, ofreciendo un acercamiento más familiar que permita al usuario brindar información de mejor calidad.

Ampliar la variedad idiomática del chatbot enfocados en países en los cuales la difusión del inglés no sea amplia, en particular algunos del sudeste asiático.

Repositorio

El código fuente del chatbot, los prompts y los conjuntos de datos que sustentan este proyecto están disponibles bajo solicitud y autorización del propietario en el siguiente vínculo: https://github.com/vladimirtamayo/tesis_macc.git

REFERENCIAS

[1] A. Vaswani y a. et, «Attention is all you need,» de *31st Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017.

[2] D. Jurafsky y J. Martin, «Transformers and Pretrained Language Models» de *Speech and Language Processing*, Upper Saddle River, NJ: Prentice Hall, 2023.

[3] D. Jurafsky y J. Martin, Speech and Language Processing, Upper Saddle River, NJ: Prentice Hall, 2023.

[4] Y. Bengio, P. Simard y P. Frasconi, «Learning Long-Term Dependencies with Gradient Descent is Difficult,» *IEEE Transactions on Neural Networks,* vol. 5, nº 2, pp. 157-166, 03 1994.

[5] T. Brown y e. al., «Language Models are Few-Shot Learners,» de *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, 2020.

[6] L. Reynolds y K. McDonell, «Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm,» 2021.

[7] V. I. Levenshtein, «Binary codes capable of correcting deletions, insertions, and reversals,» *Soviet Physics Doklady,* vol. 10, nº 8, pp. 707-710, 1966.

[8] H. M. Rojas Escobar, «Programación Dinámica,» de *Optimización no lineal y dinámica*, Bogotá, Unbiblios, 2001, p. 289.

[9] R. Wagner y M. Fischer, «The String-to-String Correction Problem,» *Journal of the ACM,* vol. 21, nº 1, pp. 168-173, 01 01 1974.

[10] A. Fawzy, «Measuring Text Similarity Using the Levenshtein Distance,» 2020. [En línea]. Available: https://blog.paperspace.com/measuring-text-similarity-using-levenshtein-distance/. [Último acceso: 18 05 2024].

[11] K. Kukich, «Techniques for Automatically Correcting Words in Text,» *ACM Computing Surveys,* vol. 24, nº 4, pp. 377-439, 01 12 1992.

[12] G. Navarro, «A Guided Tour to Approximate String Machine,» *ACM Computing Surveys,* vol. 33, nº 1, pp. 31-88, 01 03 2001.

[13] L. Tarcetti, M. Ferraro y R. Rebón, «Medium,» 29 05 2023. [En línea]. Available: https://medium.com/redbee/gpt-3-vs-gpt-4-historia-funcionamiento-y-diferencias-entre-estos-modelos-de-lenguaje-ia-9fd9214dae4c. [Último acceso: 30 05 2024].

[14] Replicate, «Run AI with an API,» San Francisco, 2024.

[15] SBERT.net, «Sentence Transformer,» 2024.

[16] B. Q. Weirui Kuang, «FederatedScope-LLM - A Comprehensive Package for Fine-tuning Large Language Models in Federated Learning,» *arXiv - Cornell University,* p. https://arxiv.org/abs/2309.00363, 2023.

[17] «LTD., Sofiri PTY,» Instant Qualified Applicant Platform (IQAP)., [En línea]. Available: https://sofiri.com/iq-connecting-student-to-advisors/. [Último acceso: 24 10 2023].

[18] M. H. Keivalya Pandya, «Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations - 3rd International Conference on “Women in Science & Technology: Creating Sustainable Career”,» *Cornell University,* p. https://arxiv.org/abs/2310.05421 , 2023.

[19] A. Paleyes, R. Urma y N. Lawrence, «Challenges in Deploying Machine Learning: a Survey of Case Studies,» *arxiv Cornell University,* p. https://arxiv.org/abs/2011.09926, 2020.

[20] S. Ozdemir, Quick Start Guide to Large Language Models - Strategies and Best Practices for using ChatGPT and Other LLMs, Addison-Wesley, 2024, p. 86.

[21] I. Goodfellow, Y. Bengio y A. Courville, «Deep learning,» *Genetic Programming and Evovable Machines,* vol. 19, p. 800, 29 Oct 2017.

[22] S. Bird, E. Klein y E. Loper, Natural Language Processing with Python, Sebastopol, CA: O'Reilly, 2009.

[23] H. Zhu y P. Koniusz, «Transductive Few-shot Learning with Prototype-based Label Propagation by Iterative Graph Refinement,» Computer Vision Foundation, 2023.

[24] International Business Machine IBM, «¿Qué es el aprendizaje few-shot?,» IBM Corporation, 2023.

[25] NLTK Project, «NLTK Documentation,» 2023.