ISSN 2007-9737

Chatbot basado en el aprendizaje profundo para recomendar productos relevantes

Luis A. Pachas-Santos, Hugo D. Calderón-Vilca,
Flor C. Cardenas-Mariño

Universidad Nacional Mayor de San Marcos, Lima,
Perú

{luis.pachas1, hcalderonv, fcardenasm}@unmsm.edu.pe

Resumen. Las plataformas de compras en línea están creciendo a un ritmo sin precedentes en todo el mundo. Estas plataformas se basan principalmente en motores de búsqueda, que todavía se basan principalmente en la base de conocimientos y utilizan palabras clave que coinciden para encontrar productos similares. Sin embargo, los clientes quieren un enfoque más interactivo que sea conveniente y confiable para consultar productos relacionados. En este artículo, proponemos una idea novedosa de buscar productos en un sistema de compra online utilizando un modelo de chatbot que utiliza Deep Learning el cual procesa imágenes y texto para generar respuestas al cliente. Un usuario puede proporcionar una imagen o dar una descripción del producto que busca, y se le presentarán productos similares basados en imágenes. El sistema de recomendación propuesto se basa en la recuperación de imágenes basadas en el contenido brindado. La evaluación del modelo propuesto se divide en dos partes: si se realiza a través de imágenes genera un 80.6% de precisión y si se realiza por texto presenta un 75% de precisión.

Keywords. Productos relevantes, chatbot, machine learning, redes neuronales, chatbot para ventas.

Chatbot based on Deep Learning for Recommending Relevant Products

Abstract. Online shopping platforms are growing at an unprecedented rate around the world. These platforms are mostly based on search engines, which are still mostly knowledge base based and use matching keywords to find similar products. However, customers want a more interactive approach that is convenient and reliable for viewing related products. In this article, we propose a novel idea of searching for products in an online shopping system using a chatbot model that uses Deep Learning which processes images and text to generate responses to the customer. A user can provide a picture or give a description of the product they are looking for, and they will be presented with similar products based on the images. The system of proposed recommendations is based on the recovery of images based on the content provided. The evaluation of the proposed model is divided into two parts: if it is done through images it generates 80.6% accuracy and if it is done through text it has 75% accuracy.

Keywords. Relevant products, recommendation, chatbot, machine learning, neural networks, selling chatbot.

1. Introducción

En la actualidad el brote del COVID-19 ha obligado al gobierno a tomar medidas de bloqueo, cerrando temporalmente empresas físicas provocando el aumento de desempleo. Las estadísticas de la ONS muestran un aumento de la tasa de desempleo de 3.8% a 4.5%. Las empresas no pueden mantener a todos sus empleados, reduciendo las ventas y limitándose en el manejo de sus recursos [1].

Una encuesta reciente realizada a 2200 adultos en los Estados Unidos, en promedio 67% de los encuestados están optando por realizar compras en línea debido a la pandemia.

Aumentando el tráfico web de los minoristas en un 16% desde el brote del COVID-19, los consumidores cada vez compran en línea más seguido acelerando un cambio estructural en las empresas.

Sin embargo, la pandemia ha afectado tanto a los consumidores como mercados recibiendo una atención limitada [2]. La digitalización es una transformación en curso de la sociedad contemporánea y abarca muchos elementos de la vida empresarial y cotidiana [3].

En la actualidad, existen diversos sistemas de chatbot para la recomendación de productos y/o servicios que ayudan a facilitar las compras, basándose en el historial de compra de los usuarios, el historial de búsqueda realizado por los clientes o en base a preguntas por la interacción del chatbot [4].

Estos sistemas pueden tomarse un tiempo hasta encontrar los productos que está buscando el cliente y este termine optando por buscarlo manualmente.

Por ello, se ha venido desarrollando varios sistemas chatbot que utilizan redes neuronales para el procesamiento de los mensajes y con ello mostrar respuestas con un tiempo más aceptable y preciso [5], no solo para la recomendación de productos sino también para brindar información adicional con respecto a las empresas o detalles de ciertos productos.

Estudios como el de Perez [6] proponen el desarrollo de un chatbot llamado ChatPy que es un agente conversacional adecuado para el uso de las pymes, este sistema utiliza redes neuronales recurrentes (RNN) y procesamiento del lenguaje natural. Pero este sistema podría mejorarse si aceptara imágenes como entrada para agilizar el proceso de recomendación.

En el caso de [7], desarrollaron un sistema chatbot que combina múltiples categorías de productos, memoriza conversaciones, genera historial de compras para la recomendación de productos.

Este sistema utiliza algoritmos de procesamiento del lenguaje natural (NLP), machine Learning y redes neuronales artificiales.

Pero el sistema necesita que el usuario haya realizado una compra o haya interactuado con el chatbot respondiendo a sus preguntas para poder realizar una recomendación de productos que se ajusten a sus necesidades, resultando una desventaja si eres un cliente nuevo.

En la investigación de [8] se desarrolló un sistema chatbot basado en lenguaje de marcado de inteligencia artificial (AIML) que se puede utilizar como un asistente de comercio electrónico. Las preguntas de entrada de los usuarios pasan por tres etapas: análisis, coincidencia de patrones y rastreo de datos utilizando AIML. Este sistema puede ser útil para preguntas comunes o mapeadas con anterioridad, pero se podrían obtener respuestas erróneas si las preguntas no coinciden con los patrones establecidos.

El estudio de [9] presenta un modelo de chatbot basado en ontologías que facilita a los clientes la obtención de información sobre las marcas, descuentos, precios, compras en línea. Las respuestas se generan haciendo coincidir las palabras clave en las consultas y recuperando las respuestas de las representaciones semánticas.

El sistema tiene buenos resultados, pero podría ser un poco tedioso representar la mayoría de los contextos para responder las preguntas.

En el caso de [10] se desarrolló un chatbot que vende bienes y servicios todo ello a través del API de Telegram que brinda servicios de Bot Messenger.

Sin embargo, este sistema se limita a integraciones con sistemas de WooCommerce para la obtención de los productos, dejando de lado diversos sistemas de ventas que no están desarrollados en PHP y WooCommerce.

Este artículo hace uso de una red neuronal tipo transformer para la implementación del modelo de respuesta. Las redes neuronales tipo transformer son una clase reciente de redes neuronales para secuencias, basadas en la autoatención [11], que han demostrado estar bien adaptadas al texto y actualmente están impulsando importantes avances en el procesamiento del lenguaje natural.

Este artículo propone el uso de un modelo chatbot que utiliza Deep Learning para recomendar productos que se ajusten a las necesidades de los clientes.

Este modelo está basado en redes neuronales tipo transformer para el análisis de texto e imágenes con la finalidad de recomendar productos a los usuarios ajustándose a las necesidades del cliente.

Este modelo se puede utilizar para la implementación de chatbots para las empresas y agilizar los procesos de venta.

2. Trabajos relacionados

2.1. Modelos basados en lenguaje de marcado y reglas

Existen diferentes tecnologías para recomendar productos. Entre ellas se encuentran los modelos basados en lenguaje de marcado y reglas como el de [8] que desarrolló un chatbot basado en la técnica de lenguaje de marcado (AIML) el cual procesa la entrada de los usuarios en tres etapas: análisis, coincidencia de patrones y rastreo de datos. Este Sistema tiene un tiempo de respuesta de 3,4 segundos. La investigación de [12] aborda el tema de las emociones en las conversaciones para establecer una conversación con el cliente y llegar al objetivo que busca la persona.

El investigador utiliza grafos que representan los flujos de una conversación, obteniendo una precisión del 63% en las respuestas. En [9] utilizan un modelo basado en ontologías, obtiene las respuestas haciendo coincidir las palabras clave en las consultas y recupera las respuestas de las representaciones semánticas. Esta investigación obtiene un 70% de exactitud, 60% de respuestas lógicas y 60% de respuestas estructuradas.

2.2. Modelos basados en lenguaje natural y machine learning

Las investigaciones basados en procesamiento del lenguaje natural y machine learning también representan una solución para este tipo de problemas, la investigación de [13] planteó un sistema chatbot distribuido que comprende varios servicios para la consulta de información. Presenta un reconocimiento del 90% de frases basados en una plantilla y una tasa de reconocimiento del 65% para consultas con sinónimos.

Un modelo que utiliza NLP para el procesamiento de texto es el de [14] el cual utiliza TensorFlow para crear un modelo neuronal para que el bot pueda capacitarse en función a un archivo de intención (dataset). El modelo obtiene un 68,51% de precisión dejando la posibilidad de mejorarlo con un dataset más variado.

Un chatbot para ventas utilizando ML (Machine Learning) es propuesto por [15], el sistema utiliza varias bibliotecas basados en Python como Spacy y Recast.ai para la implementación del chatbot. Este sistema tuvo un 50% de aprobación, 60% de los encuestados afirman que cumple con todas las funcionalidades.

2.3. Modelos basados en redes neuronales

Los modelos basados en redes neuronales son más recomendados para la implementación de chatbots por su capacidad de respuesta y precisión. En el artículo [16] planteó desarrollar un prototipo de chatbot que utilice la técnica de aprendizaje profundo para mejorar la eficiencia del chatbot, utilizó una combinación de redes neuronales de convolución (CNN) y redes neuronales recurrentes (RNN).

La investigación de Aarthi [17] desarrolló un chatbot que utiliza la búsqueda y el historial del usuario para analizar los productos que le interesan para ello utilizó un red neuronal recurrente (RNN) el cual utiliza dos componentes: un codificador y un decodificador para procesar los datos y mostrar la respuesta.

La investigación de [7] desarrolló un modelo que combina múltiples categorías de productos, memoriza conversaciones para la recomendación de productos, para ello utilizó algoritmos de procesamiento del lenguaje natural (NLP), machine learning – técnica de aprendizaje automático y redes neuronales artificiales (ANN). Este modelo está integrado con el asistente de Google para llegar a más usuarios.

Otra investigación que utiliza redes neuronales y procesamiento del lenguaje natural es de [6] el cual desarrolló un chatbot llamado ChatPy que es un agente conversacional adecuado para el uso de las pymes. Utilizaron flujos de redes neuronales recurrentes (RNN) y procesamiento del lenguaje natural, además del uso de plataformas, frameworks y SDK para el desarrollo del chatbot. Los resultados mostraron un aumento de clientes del 65.7% y ventas proporcionales al aumento de interacciones con el sistema.

Por último tenemos la investigación de [18], ellos plantearon un sistema de recomendación de productos basado en imágenes que busca de una manera más eficiente los productos en un sistema de compras en línea. El sistema cuenta con dos fases principales, la primera fase aprende o analiza la clase o tipo del producto en función a las

características del a imagen brindada por el usuario y la segunda fase propone productos similares estrechamente relacionados.

Utiliza redes neuronales de convolución (CNN) para que el modelo aprenda del conjunto de características de la imagen y pueda relacionar con otros objetos, aplicando una estructura DL-RF para la extracción y clasificación compuesta por 5 capas de convolución. Evaluaron su modelo seleccionando un conjunto de dato de 3.5 millones de productos de Amazon que constan de 20 categorías, utilizando 100 imágenes por clase.

Los resultados muestran un 75% de precisión logrando un 84% si se integra con Deep Learning (DL) en su primera fase y para la segunda fase obtuvieron 98% de recomendaciones correctas demostrando su eficiencia para la recomendación de productos.

Las investigaciones de [16] y [17] utilizan el modelo de secuencia a secuencia (Seq2Seq) en sus implementaciones, que constan en un codificador y un decodificador que procesa los mensajes. A comparación de [7] y [6] que utilizan redes neuronales artificiales (ANN) y procesamiento del lenguaje natural (NLP) para procesar los mensajes brindados por el usuario.

En el análisis de los artículos se encontró dos propuestas de arquitectura para el desarrollo de un chatbot. [6] propone una arquitectura basada en servicios, empezando a través del API de Facebook Messenger, luego Dialogflow y sus componentes, y un hosting en Heroku que utiliza Webhook. [16] también propone una arquitectura basada en servicios dividiendo el sistema en Front-end y Back-end, en el primero utiliza Mobx Store donde se almacena la lógica de la aplicación web y React para las interfaces y eventos; en el segundo utiliza TensorFlow para el análisis de mensajes, OpenCV4nodejs para comparar imágenes de productos, Socket.IO y NodeJS para la transmisión de datos y servidor respectivamente.

[17] utiliza una red neuronal recurrente de células GRU con mecanismo de atención, que contienen tres mecanismos que pueden generar una respuesta a una emoción específica. El GRU tiene como objetivo resolver el problema del gradiente de desaparición que tiene una red neuronal recurrente estándar, utilizándolo como celda básica para construir un decodificador. [16] aparte de utilizar el modelo secuencia a secuencia (Seq2Seq) al igual que [17], utiliza Word2Vec para crear un vector de palabras para organizar los datos. Las preguntas e historias se indexan y se procesan en el modelo Word2Vec.

2.4. Modelo basados en servicios API

Otros modelos que abarcan este tipo de problemas es el uso de servicios de redes sociales y chat. La investigación de [19] plantearon el desarrollo de un chatbot a través de Facebook Messenger compatible con la plataforma ManyChat para aumentar la cantidad de clientes potenciales y facilitar la información de los productos a los clientes. ManyChat es una herramienta de bajo costo que ofrece más funciones gratuitas que otras plataformas, facilita la categorización de clientes, integración con comercio electrónico y tiene la posibilidad de crear diferentes secuencias para la gestión de flujos de trabajo dando más realismo a las conversaciones.

La implementación del chatbot aumentó el porcentaje de clientes potenciales a un 25% del total de usuario que ingresaba al sitio web. En la investigación de [10] se diseñó un chatbot que pueda vender bienes y servicios a través de folletos, redes sociales, correos electrónicos, catálogo web; además de utilizar el historial de compras e información del cliente para enviar ofertas y promociones.

Para realizar dicho sistema se utilizó el API de Telegram que brinda un servicio de Bot Messenger para utilizarlo en otros sistemas, la interacción con el chatbot se realiza a través de la red social Telegram. El chatbot está escrito en PHP y utiliza MySQL para almacenar la información de los clientes y estado de los pedidos. Para brindar información de productos el chatbot necesita del identificador del producto ya que se comunicará a la API de WooCommerce para recibir la información del producto y a la vez recomendar productos complementarios.

La investigación de [20] presenta un marco unificado que utiliza UGC (contenido generado por el usuario) mediante el preprocesamiento y la alimentación de pocas palabras para determinar la validez, nombre del producto, nombre de la organización para filtrar la conversación específica de la propia organización. Su objetivo fue minar las

promociones y discusiones sobre productos de UGC, para ello se eligió SMP Twitter (plataforma de red social - Twitter) que es un excelente lugar ya que genera UGC rápidamente.

Entrenó a un codificador para aprender y asignar palabras a un vector de oración, seguido de un decodificador (modelo de red neuronal) para generar las oraciones circundantes que irán como respuesta al chatbot.

Como resultado este modelo permite mejorar la experiencia del cliente dando respuestas en vivo sobre sus consultas y preguntas. Otra modelo que utiliza el servicio de mensajería para implementar un chatbot de recomendación es el modelo de [21], implementaron un chatbot que utiliza una interfaz gráfica (GUI) que se centra en recomendar productos que se adapten a la preferencia del usuario.

Además, propusieron una estrategia de conversación donde el chatbot combina preguntas sobre preferencias y recomendaciones, a la vez que se retroalimenta por la interacción de los usuarios. Para realizar este sistema utilizaron un conjunto de datos de lugares turísticos de Kochi en Japón, también usaron el servicio de mensajería de LINE como plataforma para construir el prototipo de chatbot.

Esta plataforma brinda un API de mensajería que facilita el desarrollo de un chatbot ya que no solo permite enviar mensajes, sino que también usar una interfaz de usuario. Este artículo presentó una estrategia de conversación que intercala recomendaciones con preguntas sobre las preferencias del usuario, el cual se puede controlar configurando los valores de los parámetros.

Las investigaciones de [19] y 10] complementan sus chatbots con sistemas externos para la obtención de información o clasificación de clientes. El estudio de [19] menciona a ManyChat una plataforma que categoriza clientes además de la facilidad de integración con un comercio electrónico. En [10] plantea la integración con WooCommerce que es una plataforma de comercio electrónico para gestionar los productos y clientes de una empresa.

En [20] a diferencia de los otros investigadores, decidió utilizar data generada por Twitter para entrenar a su chatbot. Ya que esta red social contiene mucha información de promociones o discusiones sobre productos generados por el usuario. Entrenando a su codificador y decodificador para generar respuestas en vivo sobre las preguntas o consultas de los usuarios.

3. Metodología para la construcción del modelo chatbot de recomendación de productos

El modelo propuesto está basado en redes neuronales tipo transformer para el análisis de texto e imágenes con la finalidad de recomendar productos a los usuarios ajustándose a las necesidades del cliente.

3.1. Conjunto de datos

El dataset “Flickr30K” es un conjunto de datos, una base de conocimientos que conecta los conceptos de imágenes estructuradas con el lenguaje. Este dataset se utiliza para entrenar el modelo ViLBERT para identificar objetos en una imagen y mostrar una descripción de lo que se encontró.

El dataset se encuentra en inglés y para obtener las descripciones de las imágenes en español se ha usado de la API de Google para su traducción. La estructura del dataset se presenta en la tabla 1.

Para la evaluación de la recomendación de productos se utilizó un conjunto de datos de productos llamado “Flipkart”. Este conjunto de datos consta de veinte mil productos. La estructura del dataset se muestra en la tabla 2.

3.2. Arquitectura de la solución

La arquitectura propuesta para desarrollar un chatbot que automatice los procesos de venta y marketing está basado en un modelo de recomendación de productos que utiliza Deep Learning.

Esta arquitectura está constituida de 4 módulos: Módulo de Pre-Procesamiento, Módulo Procesamiento Imagen, Módulo de Entrenamiento y Módulo de Recomendación, como se muestra en la Figura 1.

El módulo de Pre-Procesamiento está conformado por el componente de Limpieza de

Tabla 1. Estructura del dataset Flickr30K

Nombre	Tipo	Descripción
image_id	Int	ID de la imagen que contiene la región
regions	Object array	Matriz de descripciones de región para esta imagen
region_id	int	ID de la descripción de la región
width	int	ancho del cuadro delimitador de la región
height	Int	altura del cuadro delimitador de la región
phrase	string	frase de descripción de la región

Tabla 2. Estructura del dataset Flipkart

Nombre	Tipo	Descripción
uniq_id	int	Identificación del producto
product_url	string	URL del producto
product_name	string	Nombre del producto
product_category	string	Categoria del producto
retail_price	float	Precio de venta
image	string	URL de la imagen

datos, que se encargará de limpiar los datos que no tengan relevancia en el mensaje de entrada como las urls, emoticones, caracteres especiales y texto en otro idioma. Luego de limpiar los datos, el mensaje se transfiere al Módulo de Entrenamiento para su procesamiento.

El módulo de Procesamiento de Imagen está formado por el componente de Proceso de Imagen el cual reducirá el tamaño de la imagen de entrada para que sea procesado con mayor rapidez.

Luego de reducir la imagen se obtendrá una región aleatoria con un tamaño específico que será enviado al Módulo de Entrenamiento donde se segmenta para poder ser procesado.

El módulo de entrenamiento está conformado de 4 componentes: Primero tenemos el componente Embedding Texto que procesa el mensaje convirtiéndolo en un vector de palabras asignando un número a cada palabra para poder ser procesado por el modelo.

El componente Embedding Imagen procesa la imagen de entrada dividiéndolo en segmentos de bits para luego ser enviado al modelo de respuesta. El componente ViLBERT será entrenado en base a los embedding de texto e imágenes para que el chatbot identifique las imágenes o texto de entrada y obtenga la respuesta más acertada dependiendo de la tarea que se le especifique.

El componente ViLBERT realizará las siguientes tareas: identificar elementos en la imagen, responder preguntas. El módulo recomendación está compuesto por el componente Productos Recomendados que se utiliza para recomendar los productos.

Una vez analizado el mensaje y encontrado la intención del usuario se procederá a recomendar productos disponibles, para ello este componente buscará los productos en base a la respuesta obtenida en el módulo anterior comparándolo con los productos almacenados en la base de datos. Por último, se clasifica los productos encontrados por categoría y relevancia para enviarlo al usuario.

3.3. Pre-procesamiento

3.3.1. Limpieza de datos

Este componente está centrado en limpiar la columna ‘phrase’ de palabras que no den un contexto entendible a la imagen de referencia. Para comenzar con la limpieza de datos se elimina espacios innecesarios, a continuación, se procede a eliminar las urls, símbolos y palabras que no sean en español ya que no aportan en la generación de respuestas.

Además, se prefiere establecer un diccionario netamente de palabras que no confundan al modelo. Luego de haber limpiado los datos se procede a verificar los datos faltantes, en caso se encuentren valores nulos se procederá a completar el campo haciendo referencia a la columna ‘synsets’ donde se encuentran sinónimos de la columna ‘phrase’.

En el peor de los casos, de no encontrar una referencia se procede a eliminar el registro.

3.4. Reducción de imagen

Este componente ajusta la imagen a un tamaño más estándar para que se agilice su procesamiento. Cambia el tamaño de la imagen al azar de manera que la longitud lateral más corta

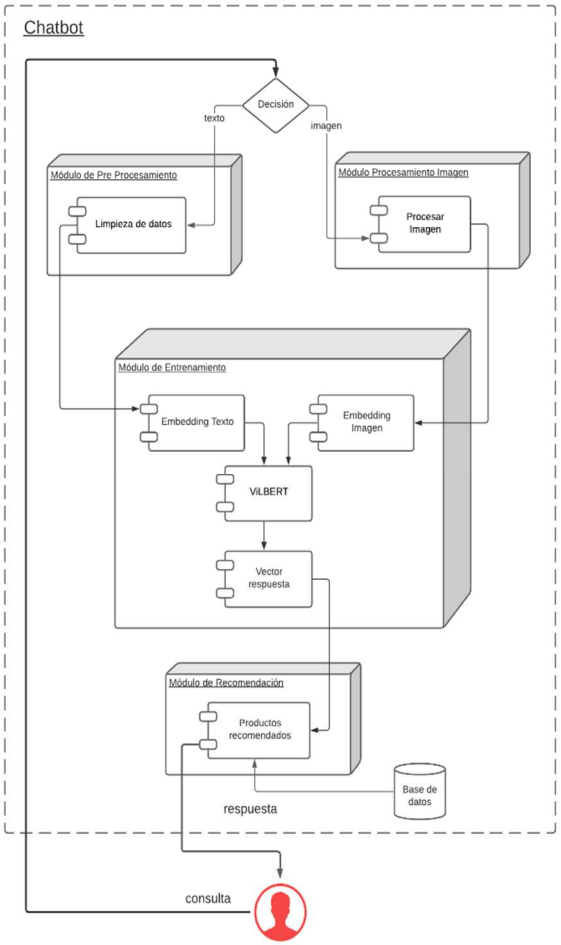

Fig. 1. Arquitectura del modelo

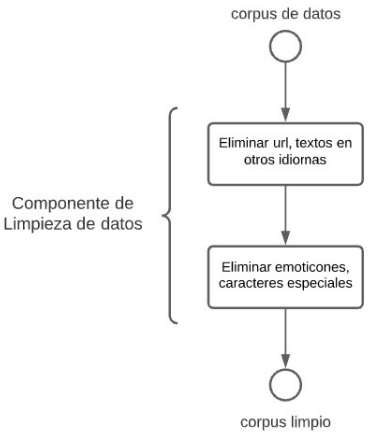

Fig. 2. Proceso de limpieza de datos

esté en el rango [256, 384]. A continuación, tomamos un recorte aleatorio de 224 x 224 que será enviado al siguiente módulo para que se divida en regiones aún más pequeñas.

Este componente utiliza el algoritmo de compresión Guetzli desarrollado por Google el cual tiene como objetivo una excelente densidad de comprensión con alta calidad visual [22]. Las imágenes generadas por Guetzli suelen ser un 20 a 30 % más pequeñas que las imágenes generadas por otras librerías como libjpeg. Esta librería solo genera archivos JPEG secuenciales debido a las velocidades de descompresión más rápidas que ofrece.

3.5. Entrenamiento

3.5.1. Embedding de texto

El componente Embedding de Texto recibe el mensaje para adaptarlo al procesamiento del modelo ViLBERT. El mensaje de entrada se tokeniza en “n” tokens de subpalabras {w0, …, wn-1} usando el método WordPiece. Los tokens especiales como [CLS] (clasificar) y [SEP] (separar), también se agregan a la secuencia de texto tokenizada. El embedding final para cada token de subpalabra se genera combinando su embedding de palabra original, embedding de segmento y embedding de posición de secuencia. Todos estos embeddings se inicializan a partir del modelo público de BERT previamente entrenado.

Usamos incrustaciones de WordPiece con un vocabulario de 30.000 tokens. El primer token de cada secuencia es siempre un token de clasificación especial ([CLS]). Los pares de oraciones se agrupan en una sola secuencia, primero los separamos con un token especial ([SEP]). En segundo lugar, agregamos una incrustación aprendida a cada token que indica si pertenece a la oración A o la oración B. Como se muestra en la Figura 3, denotamos la incrustación de entrada como E.

Por un token dado, su representación de entrada se construye sumando las incrustaciones de token, segmento y posición correspondientes, esta representación de entrada (input) se envía al componente ViLBERT para ser procesado junto al embedding de imágenes. Una visualización de esta construcción se puede ver en la Figura 3.

3.5.2. Embedding de imagen

El componente Embedding Imagen utiliza el modelo Faster R-CNN (Ren et al., 2017) que es el mejor y más rápido de los modelos de detección de objetos que introduce la Región de Propuestas Regionales (RPN) el cual predice simultáneamente los límites de los objetos y las puntuaciones de los objetos en cada detección, generando propuestas de objetos y predice la clase real del objeto.

El embedding de imágenes se genera a partir de la entrada visual mediante un proceso similar al embedding de texto. Se utiliza el modelo Faster-RCNN previamente entrenada para extraer características de "n" RoIs (regiones de interés) de una imagen, denotadas por {rn, …, rn-1} para representar su contenido visual.

El modelo Faster-RCNN obtiene la imagen que se modificó previamente para poder ser procesada en menor tiempo, luego a pasa por dos fases donde la primera consiste en obtener características de la imagen; la segunda fase genera propuestas de objetos los cuales serán puntuadas y clasificadas.

Los objetos detectados no solo pueden proporcionar contextos visuales de la imagen completa para la parte lingüística, sino que también pueden relacionarse con términos específicos a través de información detallada de la región.

También agregamos embeddings de posición a los embeddings de imágenes codificando la ubicación del objeto con respecto a la imagen global en un vector de dimensión 5: $c^{(i)} = (\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{(x_{br}-x_{tl})(y_{br}-y_{tl})}{WH})$, donde (x_{tl}, y_{tl}) y (x_{br}, y_{br}) detonan las coordenadas top-left y bottom-right del cuadro delimitador del objeto y, $\frac{(x_{br}-x_{tl})(y_{br}-y_{tl})}{WH}$ denota el área de proporción con respecto a la imagen completa.

Cada embedding se proyecta en un vector con el mismo tamaño de embedding que el tamaño oculto en las subcapas de Transformer del modelo ViLBERT [23]. Como salida de este componente tenemos al vector c que contiene las coordenadas de los objetos dentro de una imagen, este vector se envía al componente ViLBERT para que se procese junto al embedding de texto.

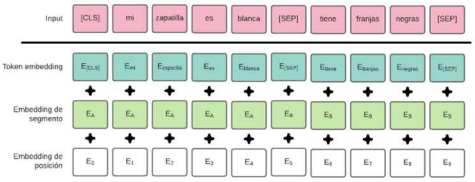

Fig. 3. Método de tokenización con WordPiece

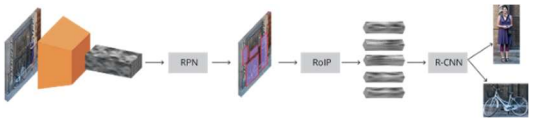

Fig. 4. Método Faster R-CNN

3.5.3. Componente ViLBERT

El componente está compuesto por el modelo ViLBERT (Visión y Lenguaje BERT), un modelo para el aprendizaje de representaciones conjuntas agnósticas de tareas de contenido de imagen y lenguaje natural. Este modelo procesa entradas visuales y textuales en flujos separados que interactúan a través de capas de transformadores de atención conjunta. El componente ViLBERT recibe los embedding de texto y de imagen de los módulos anteriores para calcular las correspondencias entre las regiones de imagen y texto agregando las capas que hacen este cálculo con co-atención.

La entrada al modelo ViLBERT se representa por e (i) para cada imagen RoI (regiones de interés) r(i), combinando el embedding de imagen y el embedding de texto.

Para ello se suman los embedding de los objetos, embedding de segmentos, embedding de posición de imagen y el embedding de posición de secuencia realizados en los módulos anteriores, una representación de este proceso se puede ver en la Figura 5.

Los módulos de co-atención que están dentro del componente ViLBERT revisarán las otras ramas de flujo para calcular las puntuaciones importantes entre las imágenes – texto y viceversa.

Estos módulos calculan la atención en función del texto y las imágenes, luego se agrega una capa de módulo transformador y otra capa de co-atención para mejorar la precisión en la relación, mientras más capas se aumente se reflejará un aprendizaje más profundo. La salida dependerá de

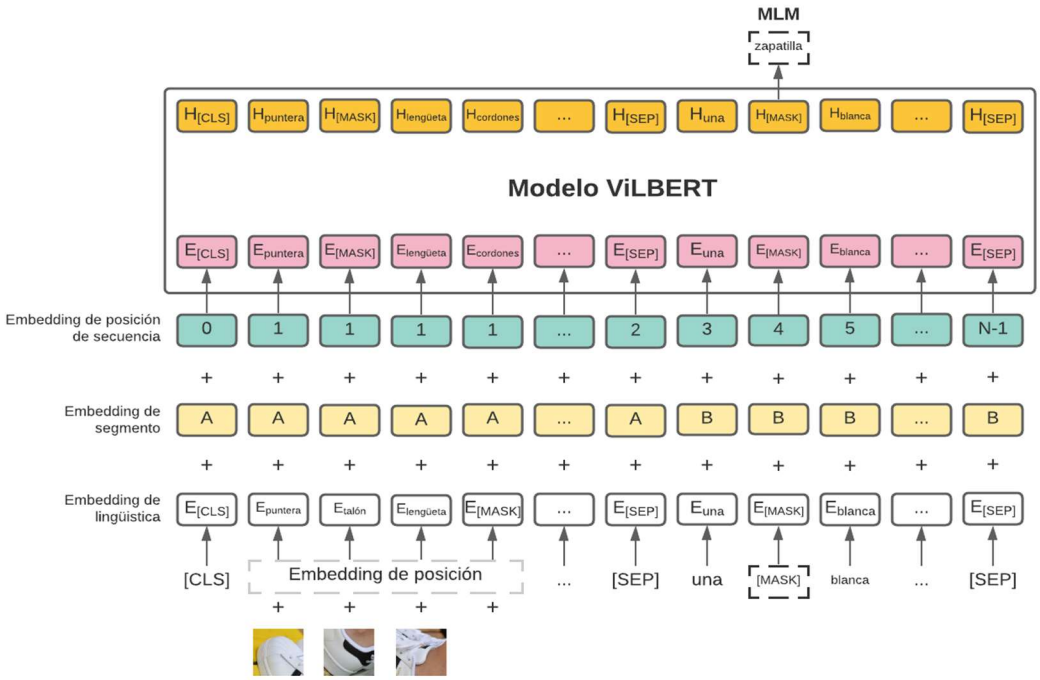

Fig. 5. Arquitectura del modelo ViLBERT

la tarea que se le asigne al procesar la data de entrada.

El modelo se entrenó para realizar la tarea MLM (Masked Language Modeling) en base al dataset de imágenes. MLM dará como resultado la clasificación de la imagen a nivel global, un ejemplo de esta tarea se ve reflejado en la Figura 5 el cual identifica los objetos talón, lengüeta, puntera, cordones y predice la clasificación de todos los objetos dando como resultado una 'zapatilla'.

3.5.4. Componente vector respuesta

Este componente consiste en decodificar la respuesta obtenida del modelo ViLBERT para poder entenderlo. En base a las puntuaciones se buscarán las palabras que coincidan con los números obtenidos, estas palabras ya se encuentran definidas en una biblioteca de palabras que contiene el modelo BERT. Se arma la respuesta con los resultados obtenidos producto de la decodificación y se enviará al módulo de recomendación para ser procesado.

3.6. Módulo de recomendación

El componente de recomendación de productos obtiene la respuesta del módulo anterior y pasará a buscar coincidencias de texto en base a los productos guardados en la base de datos. Buscando coincidencias en los campos: nombre, descripción, categoría, subcategorías.

Luego de obtener los productos relacionados a la búsqueda se clasificará los productos por categorías y mayor relevancia para ser enviado al usuario. Se enviará la imagen, descripción y precio del producto para que el usuario pueda elegir el producto que más se acerque a lo que busca.

4. Resultados

Los resultados del modelo propuesto se pueden resumir de la siguiente manera: El sistema puede buscar productos con imágenes o texto ingresado por el usuario haciendo posible una mayor interacción con el sistema chatbot, agilizando los procesos de búsqueda para el

cliente. El sistema puede almacenar el nombre y la información del cliente cuando se produce una conversación.

4.1. Preprocesamiento y evaluación del modelo basado en texto

Antes de realizar el entramiento por texto se realizó una limpieza en el conjunto de datos para eliminar campos vacíos, caracteres y símbolos innecesarios, urls. El conjunto de datos cuenta con más de 3 millones de tweets y respuestas de las marcas más importantes en Twitter.

Para realizar el preprocesamiento y posteriormente el entrenamiento basado en texto se utilizaron 200 mil registros extraídos al azar. Los resultados del preprocesamiento se pueden apreciar en la tabla 3.

Una vez realizado la limpieza de datos nos quedamos con un total de 194,400 registros listos para pasar a la fase de entrenamiento por texto. La evaluación del modelo basado en preguntas y respuestas con respecto a productos y búsqueda de información relacionadas a la empresa se muestran en la tabla 4.

Para esta evaluación recurrimos a 3 métricas: precisión, relación, dinámico. La precisión se utilizó para saber qué tan precisos son las respuestas relacionadas a las preguntas, la relación para comprobar si las respuestas se relacionan con las consultas y lo dinámico para corroborar cuantas respuestas son diferentes si el cliente hace la misma pregunta repetidamente.

4.2. Evaluación del modelo basado en imágenes

El sistema acepta los siguientes formatos de imagen: JPG, PNG, JPEG para ser procesados e identificar el objeto de búsqueda. A continuación, se muestra una tabla de resultados en el cual el modelo propuesto pudo reconocer los objetos dentro de las imágenes.

El modelo se entrenó con el dataset “Flickr30K” que consta de treinta mil imágenes de los cuales solo se utilizó mil imagenes para realizar el entrenamiento. El dataset se encuentra disponible en [24].

El modelo se entrenó con mil imágenes de los cuales el 70% se utilizó para el entrenamiento y el 30% para las pruebas. Los resultados del reconocimiento de objetos dentro de las imágenes se muestran en la tabla 5.

Tabla 3. Resultados del preprocesamiento de texto

Tipo de limpieza	% de registros
Espacios en blanco	7.65%
Caracteres y simbolos	4.5%
URL’s	15.2%
Inconsistencia	2.8%

Tabla 4. Resultados de la evaluación basado en texto

Métrica	Resultado
Precisión	79.3%
Relación	74%
Dinámico	24.6%

Tabla 5. Resultados del reconocimiento de objetos

Medición	Resultado
Precisión de predicciones de objetos.	75.83%
Porcentaje de predicción de casos positivos	78%

La evaluación del modelo con respecto a la detección de objetos arrojó una precisión del 75.8% pudiendo mejorar si se entrena con imágenes más variadas o un dataset que contenga más categorías de imágenes y diferentes ángulos de los objetos para que el modelo pueda reconocer con mayor precisión.

El reconocimiento de objetos es importante para saber que producto está buscando el cliente, así el chatbot podrá sugerir con mayor precisión algún producto. Un ejemplo de reconocimiento de objetos se puede apreciar en la Figura 6.

4.3. Evaluación de la recomendación

Para la evaluación de la recomendación de productos se utilizó un conjunto de datos de productos llamado “Flipkart” el cual se puede encontrar en [25]. Este conjunto de datos consta de veinte mil productos de los cuales se ha utilizado 4 mil registros para realizar la prueba de recomendación.

Luego de realizar el procesamiento de la imagen o texto se obtiene una descripción del producto a buscar con el cual se realiza la búsqueda del producto dentro de la base de datos. Se hizo la prueba planteando la búsqueda de 120

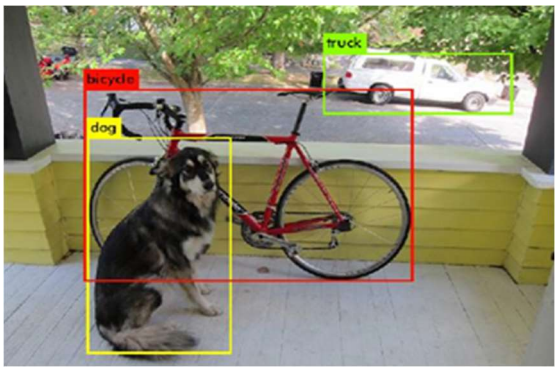

Fig. 6. Reconocimiento de objetos

productos a través del modelo obteniendo un 80.6% de precisión en la recomendación de productos.

4.4. Discusiones

Analizando los resultados podemos notar una diferencia con los artículos revisados con anterioridad. En el estudio de [18] se obtuvo una precisión de 84% una mayor precisión a diferencia de nuestros resultados en la precisión que es de 80.6% pero su trabajo se basa solo en imágenes a diferencia de este artículo que utiliza imagen o texto para la recomendación.

Nuestro modelo se adecua a diferentes plataformas ya que se podrá utilizar como un servicio y aprovechar la mayoría de plataformas a diferencia de [10] que solo está pensando en sistemas basados en php o solo sistemas como el de [19] que está basado en un API de Facebook para realizar sus funciones.

La ventaja de nuestro sistema radica en que puede interactuar de una manera más fluida con los clientes relacionando palabras y adecuándose al contexto de la conversación a diferencia de [8] que utiliza patrones para procesar las respuestas, [9] que hace coincidir las preguntas con palabras clave los cuales no podrían obtener una respuesta adecuada a ciertas preguntas.

Los sistemas de recomendación se aplican en diferentes campos como es la recomendación de videojuegos [26], por otro lado, los estudios de procesamiento de imágenes ha avanzado en el reconocimiento rostros, personalidad [27], actualmente el procesamiento de imágenes se aplica para recomendaciones de productos, así como lo hemos descrito en este trabajo.

5. Conclusiones

El presente artículo presenta un modelo de chatbot que utiliza Deep Learning para la recomendación de productos, este modelo es capaz de procesar imagen o texto para la recomendación. Presenta una novedosa red neuronal llamada.

Transformer la cual es una clase reciente de redes neuronales para secuencias, basadas en la autoatención [10], que han demostrado estar bien adaptadas al texto y también para la relación de imágenes.

Los resultados obtenidos en la evaluación del modelo muestran un 80.6% de precisión en la recomendación de productos el cual se puede mejorar con conjunto de datos más variados.

Para el caso del entrenamiento por imágenes buscar un conjunto de datos que contenga una variedad de categorías y ángulos de imágenes para mejorar la precisión.

Referencias

1. **Junlan, Z., Chengke, Y. (2020).** Data-analysis-based discussion on COVID-19 pandemic shocks to the economy and policy responses: Cases in the United Kingdom. 2020 Management Science Informatization and Economic Innovation Development Conference (MSIEID), pp. 538–541. DOI: 10.1109/MSIEID52046.2020.00109.
2. **Kim, R. Y. (2020).** The Impact of COVID-19 on consumers: Preparing for digital sales. IEEE Engineering Management Review, Vol. 48, No. 3, pp. 212–218. DOI: 10.1109/EMR.2020.2990115.
3. **Hagberg, J., Sundstrom, M., Egels-Zandén, N. (2016).** The digitalization of retailing: an exploratory framework. International Journal of Retail & Distribution Management, Vol. 44, No. 7, pp. 694–712. DOI: 10.1108/IJRDM-09-2015-0140.
4. **Ikemoto, Y., Asawavetvutt, V., Kuwabara, K., Huang, H. H. (2018).** Conversation

strategy of a chatbot for interactive recommendations. Intelligent Information and Database Systems, Vol. 1, pp. 117–126. DOI: 10.1007/978-3-319-75417-8.

5. **Anil, D., Vembar, A., Hiriyannaiah, S., Gm, S., Srinivasa, K. G. (2018).** Performance analysis of deep learning architectures for recommendation systems. 2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW), pp. 129–136. DOI: 10.1109/HiPCW.201 8.8634192.
6. **Perez, P., De-La-Cruz, F., Guerron, X., Conrado, G., Quiroz-Palma, P., Molina, W. (2019).** ChatPy: conversational agent for SMEs: A case study. Iber. Conf. Inf. Syst. Technol, CISTI, pp. 19–22. DOI: 10.23919/CIS TI.2019.8760624.
7. **Shafi, P. M., Jawalkar, G. S., Kadam, M. A., Ambawale, R. R., Bankar, S. V. (2020).** AI—assisted chatbot for e-commerce to address selection of products from multiple products. Stud. In: Dey, N., Mahalle, P., Shafi, P., Kimabahune, V., Hassanien, A. (eds) Internet of Things, Smart Computing and Technology: A Roadmap Ahead. Studies in Systems, Decision and Contro, Vol. 266, pp. 57–80. DOI: 10.1007/978-3-030-39047-1_3.
8. **Nursetyo, A., De Rosal-Ignatius, M. S., Subhiyakto, E. R. (2018).** Smart chatbot system for E-commerce assistance based on AIML 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 641–645. DOI: 10.11 09/ISRITI.2018.8864349.
9. **Nazir, A., Khan, M. Y., Ahmed, T., Jami, S. I., Wasi, S. (2019).** A novel approach for ontology-driven information retrieving chatbot for fashion brands. International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 10, No. 9, pp. 546–552. DOI: 10.14569/ijacsa.201 9.0100972.
10. **Amir-Reza, A., Hemadi, R. (2018).** Design and implementation of a chatbot for e-commerce. pp. 1–10.
11. **Coltman, J. W. (2002).** The transformer [historical overview]. In IEEE Industry Applications Magazine, Vol. 8, No. 1, pp. 8–15. DOI: 10.1109/2943.974352.
12. **Fiddin Al Islami, M. T., Ridho-Barakbah, A., Harsono, T. (2020).** Interactive applied graph chatbot with semantic recognition. 2020 International Electronics Symposium (IES), pp. 557–564. DOI: 10.1109/IES50839.2020.9 231678.
13. **Angelov, S., Lazarova, M. (2019).** E-commerce distributed chatbot system. BCI'19: Proceedings of the 9th Balkan Conference on Informatics, No. 8, pp. 1–8. DOI: 10.1145/3351 556.3351587.
14. **Singh, R., Paste, M., Shinde, N., Patel, H., Mishra, N. (2018).** Chatbot using tensorflow for small businesses. Proceedings Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1614–1619. DOI: 10.1109/ICICCT.2018.8472998.
15. **Oguntosin V., Olomo, A. (2021).** Development of an E-Commerce chatbot for a university shopping mall. Applied. Computational. Intelligence and Soft Computing, Vol. 2021. DOI: 10.1155/2021/6 630326.
16. **Prasomphan, S. (2019).** Improvement of chatbot in trading system for SMEs by using deep neural network. 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 517–522. DOI: 10.1109/ICCCBD A.2019.8725745.
17. **Aarthi, N. G., Keerthana, G., Pavithra, A., Pavithra K. (2020).** Chatbot for retail shop evaluation. International Journal of Computer Science and Mobile Computing, Vol. 9, No. 3, pp. 69–77.
18. **Ullah, F., Zhang, B., Khan, R. U. (2020).** Image-based service recommendation system: A JPEG-coefficient RFs approach. IEEE Access, Vol. 8, pp. 3308–3318. DOI:10.1109/ACCESS.2019.2962315.
19. **Illescas-Manzano, M. D., López, N. V., González, N. A., Rodríguez, C. C. (2021).** Implementation of chatbot in online commerce, and open innovation. Journal of Open Innovation: Technology, Market and Complexity (2021). Vol. 7, No. 2, pp. 125. DOI: 10.3390/joitmc7020125.

20. **Kushwaha, A. K., Kar, A. K. (2020).** Language model-driven chatbot for business to address marketing and selection of products. In: Sharma, S.K., Dwivedi, Y. K., Metri, B., Rana, N. P. (eds) Re-imagining Diffusion and Adoption of Information Technology and Systems: A Continuing Conversation, TDIT 2020, IFIP Advances in Information and Communication Technology, Springer, Vol. 617, pp. 16–28. DOI: 10.100 7/978-3-030-64849-7_3.
21. **Ikemoto, Y., Asawavetvutt, V., Kuwabara, K., Huang, H. H. (2019).** Tuning a conversation strategy for interactive recommendations in a chatbot setting. Journal of Information and Telecommunication, Vol. 3, No. 2, pp. 180–195. DOI: 10.1080/2475183 9.2018.1544818.
22. **Ren, S., He, K., Girshick, R., Sun, J. (2017)**. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 6, pp. 1137–1149. DOI: 10.1109/TPAMI.2016.2 577031.
23. **Lu, J., Batra, D., Parikh, D., Lee, S. (2019).** ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Advances in Neural Information Processing Systems, Vol. 32, pp. 1–11.
24. **Hsankesara (2018).** Flickr Image dataset | Kaggle. https://www.kaggle.com/hsankesar a/flickrimage-dataset (accessed Nov. 26, 2021).
25. **PromptCloud (2017).** Flipkart Products | Kaggle. https://www.kaggle.com/Prompt CloudHQ/ flipkart-products (accessed Nov. 26, 2021).
26. **Calderon-Vilca, H., Chavez, N. M., Guimarey, J. M. R. (2020).** Recommendation of videogames with fuzzy logic. In 2020 27th Conference of Open Innovations Association (FRUCT), pp. 27–37. DOI: 10.23919/FRUCT49677.2020.9211082.
27. **Lizama, G. B., Calderón-Vilca, H. D. (2022).** Model for automatic detection of the big five personality traits through facial images. International Journal of Computer Information Systems and Industrial Management Applications, Vol. 4, pp. 60–67.

Article received on 31/12/2021; accepted on 06/11/2022.
Corresponding author is Hugo D. Calderón-Vilca.