Arquitectura de Red Neuronal para el Desarrollo de Agentes Conversacionales destinados a la Atención al Cliente en las Redes Sociales

(Neural Network Architecture for Development of Conversational Agents for Customer Service in Social Networks)

Leonardo Javier Ibáñez[1]

Campo temático: Machine Learning.

Resumen

La atención al cliente es un tema importante para las empresas y cada vez los usuarios son más exigentes con el tiempo de respuesta, la cantidad de interacciones y la calidad de las respuestas. Para brindar dicho servicio, las empresas utilizan chatbots porque proporcionan atención las 24 horas y reducen los costos de facturación, pero carecen de flexibilidad para desarrollar diálogos reales porque enfrentan dificultades para comprender el estilo de escritura y vocabulario de los usuarios.

En este trabajo, se presenta una arquitectura híbrida combinando los modelos de conversación basados en recuperación y en generación para resolver dicha problemática y para demostrar la viabilidad del enfoque propuesto se comparó distintos algoritmos de aprendizaje profundo.

Palabras Claves: Aprendizaje Profundo; Procesamiento de Lenguaje Natural; Agente Conversacional; Servicio al Cliente.

[1] Comisión Nacional de Energía Atómica. leoo.davinci@gmail.com

Abstract

Customer service is an important issue for companies and users are increasingly demanding with the response time, the number of interactions and the quality of the responses. To provide such a service, companies use chatbots because they provide 24-hour service and reduce billing costs, but they lack the flexibility to develop real dialogues because they face difficulties in understanding the writing style and vocabulary of users.

In this work, a hybrid architecture is presented combining recovery-based and generation-based conversation models to solve said problem and to demonstrate the viability of the proposed approach, different deep learning algorithms were compared.

Keywords: Deep Learning; Natural Language Processing; Conversational Agent; Customer Service.

1. Introducción

A muchas empresas les interesa invertir en un equipo de atención al cliente disponible 24x7x365 pero a menudo esto no es realista. Los chatbots son una solución innovadora para cerrar la brecha y, al mismo tiempo, ofrecer una experiencia personalizada y en tiempo real que los usuarios desean (Slater, 2018).

Los chatbots se están convirtiendo rápidamente en una herramienta esencial para la atención al cliente y evolucionan constantemente con la integración de la inteligencia artificial porque tienen la capacidad de brindar un servicio instantáneo proporcionando respuestas rápidas sin la necesidad de pausas (Ferraro y Restrepo, 2019; Reddy, 2017; Vishnoi, 2019). Las empresas se benefician reduciendo los costos de facturación y tiempo al no tener que atender casos repetitivos o similares, permitiendo atender a cientos de clientes a la vez y dejando más tiempo para resolver problemas complejos o únicos de los clientes (Reddy, 2017).

Sin embargo, éstos presentan problemas con la calidad del servicio porque poseen limitaciones al no poder desarrollar diálogos reales debido a que enfrentan dificultades para comprender el estilo de escritura y el vocabulario de los usuarios.

Existen dos modelos de conversación para generar una respuesta y diferentes algoritmos para desarrollar dichos modelos. Sin embargo, como no se sigue un patrón de conocimiento en el desarrollo de chatbot, crear un buen chatbot sigue siendo uno de los desafíos más difíciles en el campo de la inteligencia artificial (Villar, Rodríguez y Rocha, 2018).

Este trabajo presenta un chatbot denominado A.V.I. (Agente Virtual Inteligente) con una arquitectura híbrida con la capacidad de mantener una conversación natural con un ser humano permitiendo recibir consultas de diferentes maneras y devolver respuestas consistentes. Para evaluar el chatbot, se realizó variaciones de la arquitectura de la red neuronal artificial propuesta y se utilizó métricas de desempeño para comparar los resultados obtenidos y medir la precisión de las respuestas generadas.

El resto del trabajo se organiza de la siguiente manera. La sección II describe el marco teórico. La sección III relata los trabajos relacionados. La sección IV presenta el enfoque propuesto. La sección V reporta los resultados experimentales. Finalmente, la sección VI concluye el trabajo e identifica futuras líneas de trabajo.

2. Marco Teórico

La inteligencia artificial, el aprendizaje automático y el aprendizaje profundo son parte de la razón por la cual el programa AlphaGo de Google DeepMind derrotó al maestro surcoreano Lee Se-dol en el juego de mesa Go pero no son lo mismo. La manera más fácil de pensar en su relación es visualizándola como círculos

concéntricos con la inteligencia artificial, la idea que vino primero, la más grande, luego el aprendizaje automático y finalmente el aprendizaje profundo dentro de ambos.

El aprendizaje profundo se centra en un subconjunto de herramientas y técnicas del aprendizaje automático, y los aplica a la solución de casi cualquier problema que requiera "pensar". El concepto de aprendizaje profundo a veces se conoce como "redes neuronales profundas" en referencia a las muchas capas involucradas.

Las redes neuronales artificiales a diferencia de un cerebro biológico donde cualquier neurona puede conectarse a cualquier otra neurona dentro de una cierta distancia física, estas redes neuronales artificiales tienen capas discretas, conexiones y direcciones de propagación de datos. Hay varios tipos de redes neuronales artificiales y cada uno de ellos se implementa en función de las operaciones matemáticas y del conjunto de parámetros necesarios para determinar la salida.

El desarrollo de las redes neuronales artificiales ha sido clave para enseñar a las máquinas a pensar con las ventajas innatas que ya tienen sobre nosotros como la velocidad y la precisión. Basado en los datos que se le suministran pueden hacer afirmaciones, decisiones o predicciones con cierto grado de certeza e involucran un circuito de retroalimentación para el "aprendizaje" para averiguar si sus decisiones fueron correctas o no, y luego cambiar su enfoque para mejorar la próxima vez.

Existen diferentes técnicas de inteligencia artificial aplicada en los chatbots:

2.1 Procesamiento de Lenguaje Natural

El procesamiento de lenguaje natural es el área que se enfoca en el desarrollo de sistemas que permiten la comunicación entre un ser humano y una máquina a través del lenguaje natural.

2.2 Modelo de Recuperación

Este modelo requiere previamente que un programador escriba algunas reglas y patrones para el análisis y la descomposición de la oración de entrada. El modelo explora las palabras claves y recupera las respuestas relevantes según la cadena de consulta.

Alicebot es uno de los mejores chatbots basados en recuperación (Serban, et al., 2017; Bhagwat, 2018) conformando por AIML (Artificial Intelligence Markup Language) que es una forma de XML (Extensible Markup Language) en el cual se definen las reglas para coordinar patrones y decidir respuestas, pero las desventajas de los chatbots basados en AIML es que necesitan más de diez mil patrones antes de que comiencen a percibirse realista, fallan cuando encuentran una oración que no contiene ningún patrón conocido y requieren mucho esfuerzo escribir las reglas manualmente.

2.3 Modelo de Generación

Gracias a los últimos avances en las redes neuronales artificiales es posible crear modelos de chatbots sin escribir reglas de antemano. Este modelo está capacitado para generar respuestas palabra por palabra sin intervención humana, sin embargo, es propenso a cometer errores gramaticales pero puede responder con oraciones más naturales.

A su vez, existen diferentes tipos de redes neuronales artificiales para el procesamiento de lenguaje natural:

2.4 LSTM (Long Short Term Memory)

Las redes neuronales LSTM están diseñados para recordar información durante largos períodos de tiempo.

2.5 GRU (Gated Recurrent Unit)

Las redes neuronales GRU es una variación del anterior y se utilizan cuando se necesita entrenar más rápido y no se posee mucha capacidad de cálculo a la mano.

2.6 Redes Neuronales Bidireccionales

Las redes anteriores aprenden del pasado para predecir el futuro, pero a veces se tiene que aprender de representaciones de pasos futuros para comprender mejor el contexto.

2.7 Modelo Seq2Seq (Sequence to Sequence)

El modelo Seq2Seq toma una secuencia de elementos y genera otra secuencia de elementos. Hoy en día se lo utiliza para una variedad de aplicaciones diferentes como traducción automática, subtítulos de video, etc. El modelo está conformado por dos LSTM o GRU, en el cual, uno hace de codificador y el otro de decodificador y es por eso que a veces se lo denomina red codificador-decodificador.

2.8 Técnica “Atención”

La técnica de atención permite que el modelo se centre en diferentes partes de la secuencia de entrada en cada etapa de la secuencia de salida, preservardo el contexto de principio a fin.

3. Trabajos Relacionados

A la hora de desarrollar un agente conversacional o chatbot y que aprenda automáticamente de conversaciones existentes, existen dos modelos para generar una respuesta: basado en reglas o recuperación y basado en generación; pero recientemente ha habido trabajos que combinaron dichos modelos pero no siguen un patrón de conocimiento en el desarrollo de chatbot debido a que la cantidad de publicaciones que tienen como objetivo difundir el conocimiento es baja (Villar, Rodríguez y Rocha, 2018). En una ventana de tiempo entre el 2016 y 2019 solamente hay seis trabajos enfocados en promover resultados.

Para el modelo de recuperación, en (Serban, et al., 2017) propusieron bolsa de palabras o Alicebot como en (Bhagwat, 2018) o modelo de Seq2Seq con variables Gaussianas entrenadas como autoencoders variacionales. En (Song, et al., 2016) los autores coinciden con la bolsa de palabras eliminando las palabras vacías. Por otro lado, en (Le, Nguyen y Nguyen, 2018) los autores propusieron el modelo Manhattan LSTM para conocer la similitud entre dos mensajes y en (Yang, et al., 2019) propusieron un enfoque de coincidencia contexto-contexto.

Para el modelo de generación, en (Song, et al., 2016) los autores propusieron BiSeq2Seq (Bidirectional Sequence to Sequence) utilizando RNN (Recurrent Neural Network) con GRU para que la respuesta generada sea fluida y lógica con respecto a la consulta. Los autores en (Xu, et al., 2017) propusieron algo similar en su modelo y utilizaron dos capas de LSTM donde una capa es un codificador que asigna una secuencia de entrada de longitud variable a un vector de longitud fija, y la otra capa es un decodificador que asigna el vector a una secuencia de salida de longitud variable. En (Le, Nguyen y Nguyen, 2018) también utilizaron la arquitectura codificador-decodificador. Los autores (Yang, et al., 2019; Csáky, 2017; Nuez, 2018) utilizaron un modelo Seq2Seq con atención, conocido como arquitectura transformadora, pero en (Csáky, 2017) el autor utiliza un codificador bidireccional. Por otro lado, los autores en (Bhagwat, 2018; Tiha, 2018) propusieron LSTM bidireccional con atención pero en (Bhagwat, 2018) le agrega una capa LSTM adicional. En (Serban, et al., 2017) propusieron un modelo con dos capas de GRU con una capa de salida softmax.

En (Peters, 2017-2018) el autor demostró que el modelo GRU se entrena mucho más rápido que el modelo LSTM y que el modelo GRU con secuencia de entrada invertida proporciona más precisión que el modelo LSTM.

Pero (Song, et al., 2016; Yang, et al., 2019) a diferencia del resto, incorporaron un tercer módulo el cual devuelve la respuesta final calificando candidatos en los módulos anteriores. Para esto, en (Song, et al., 2016) los autores propusieron el mismo módulo de recuperación y por otro lado, en (Yang, et al., 2019) a través de etiquetas utilizando DMN-PRF (Deep Matching Networks via Pseudo-Relevance Feedback).

Un dato importante a aclarar, es que la mayoría de los trabajos investigados utilizaron como capacidad de cálculo GPU NVIDIA, RAM de 16GB, servidores AWS (Amazon Web Services) y como conjunto de datos subtítulos de películas para entrenar la red neuronal artificial.

4. Enfoque Propuesto

La solución factible para lograr un agente conversacional inteligente es utilizando de forma conjunta las ventajas de los modelos de conversación basado en recuperación y en generación para obtener mejores resultados finales y reducir el tiempo de la red neuronal artificial sin la necesidad de utilizar hardware dedicado.

La arquitectura de A.V.I. está compuesta por tres módulos y su base de datos. La Figura 1 muestra un esquema conceptual de la arquitectura y se puede identificar claramente cómo se relacionan dichos módulos para generar la respuesta final.

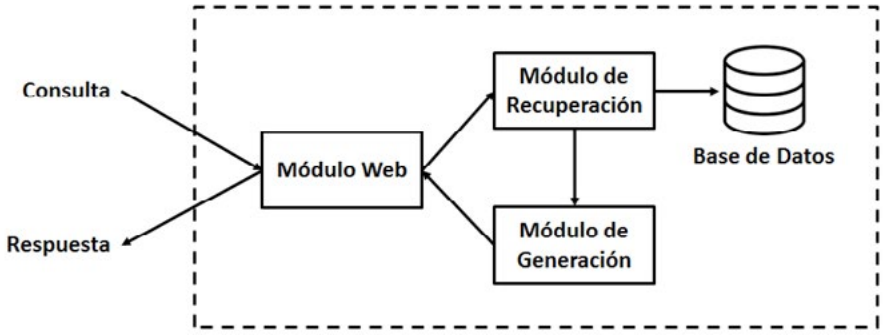

Figura 1 Arquitectura de A.V.I.(Agente Virtual Inteligente)

4.1 Módulo Web

El módulo web se desarrolló con la librería Tweepy para utilizar la API de Twitter. El chatbot A.V.I. busca periódicamente tweets en los que se le menciona y extrae el id, el autor y el texto del tweet.

4.2 Módulo de Recuperación

Cuando el chatbot A.V.I. recibe una consulta Q, el módulo de recuperación adopta un enfoque de coincidencia de contexto para buscar pares de consulta-respuesta a partir de la base de datos con conversaciones históricas y devolver un conjunto de respuestas candidatas Rc.

Para lograr esta parte se utilizó el algoritmo BM25 como motor de búsqueda para tener un modelo de recuperación simple pero eficiente.

4.3 Base de datos

La red social Twitter posee un gran conjunto de datos de conversaciones entre consumidores y agentes de atención al cliente que puede ayudar en la comprensión del lenguaje natural y los modelos de conversación.

A partir del conjunto de datos "Customer Support on Twitter"[2] obtenido del repositorio Kaggle con alrededor de 3.000.000 de tweets se hizo un proceso de ETL (Extract, Transform and Load) para formatear y limpiar los datos.

4.3.1 Extracción

Utilizando la librería pycld2 se extrajo los tweets en idioma inglés de una empresa.

4.3.2 Transformación

Aplicando técnicas de NLP y utilizando la librería NLTK se estandarizó los textos, convirtiendo todas las palabras en minúsculas, removiendo los caracteres de puntuación y las palabras vacías para reducir la posibilidad de confusión lingüística del algoritmo y se transformaron las palabras a su forma básica o raíz a través de la lematización. El proceso de lematización es lento pero es mejor que la derivación y tiene más precisión.

4.3.3 Carga

Como resultado final, se obtiene un nuevo conjunto de datos de dos columnas consulta-respuesta

Se eligió dicho conjunto de datos con conversaciones de la vida real porque es más realista en vez de utilizar subtítulos de películas ya que estos no imitan la interacción humana real.

4.4. Módulo de Generación

Luego que el módulo anterior generó un conjunto de respuestas candidatas Rc, el módulo de generación lo toma como entrada para devolver la respuesta final Rf significativa y natural.

Para lograr esta parte se utilizó aprendizaje profundo. La arquitectura de la red neuronal artificial está conformada por seis capas. La primera y segunda capa, es un modelo GRU bidireccional y con atención para no generar oraciones cortas con poca información y la cuarta capa es un modelo GRU. La Figura 2 muestra la arquitectura de la red neuronal artificial descrita.

[2] https://www.kaggle.com/thoughtvector/customer-support-on-twitter

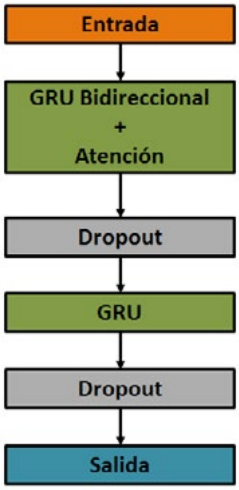

Figura 2 Arquitectura utilizada en el Módulo de Generación

Antes y después de la cuarta capa, se agregó una capa de tipo Dropout del 20%, que por lo general se utiliza dicho valor, para verificar el sobreajuste. La última capa, la capa de salida, es de tipo Dense la cual permite realizar la clasificación a partir de los datos generados por las capas anteriores a ella. La capa de atención posee una función de activación sigmoid y la capa de salida posee una función de activación softmax, porque dichas funciones de activación se aplican para transformar los valores de salida en valores de probabilidad.

El paso previo a construir la arquitectura es el procesamiento de los datos porque las palabras en las respuestas candidatas no se pueden usar directamente como entrada en un modelo de aprendizaje automático debido a que no puede comprenderlas pero si puede interpretar una serie de números. Existen dos formas de abordar esta tarea para codificar los datos y hacer predicciones, un mapeo a nivel de las palabras o al nivel de los caracteres.

Para convertir los datos de texto obtenidos del modelo de recuperación se creó un mapeo de palabras porque tiene una precisión mayor, puede preservar más fácilmente el contexto entre palabras y no requiere construir una red neuronal artificial grande. Luego, para eliminar cualquier relación ordinal que pueda haberse introducido durante el proceso de mapeo, se hizo una codificación one-hot que es un método para representar texto como una matriz binaria.

5. Resultados Experimentales

Para evaluar el enfoque propuesto se realizó una comparación entre tres variaciones de la arquitectura utilizada en la red neuronal artificial y por cada

variación, se realizó varios entrenamientos modificando la cantidad de epoch con un batch_size fijo y la cantidad de unidades en cada capa porque están relacionados con los recursos y la velocidad del proceso de entrenamiento de la red neuronal artificial con el objetivo de medir la precisión de las respuestas generadas, la curva de aprendizaje, el tiempo de entrenamiento y la precisión del modelo propuesto y determinar que variación del modelo podría ser la más óptima.

5.1 Setup del Experimento

Para el experimento se diseñaron tres modelos (A, B y C) de arquitectura de redes neuronales artificiales. El modelo A es el chatbot A.V.I. y los modelos B y C fueron utilizados como baseline en este trabajo. El modelo B es un modelo simple que se utilizó en el trabajo (Serban, et al., 2017) y está conformado por dos capas GRU. El modelo C, para explorar otras técnicas compatibles para el chatbot y observar sus resultados, se utilizó el modelo Seq2Seq. Esta técnica ha sido reportada en varios trabajos (Song, et al., 2016; Le, Nguyen y Nguyen, 2018; Yang, et al., 2019; Xu, et al., 2017; Csáky, 2017) y está conformado de una capa GRU como codificador y una capa GRU como decodificador.

Para entrenar los tres modelos se utilizó la técnica repeated random sub-sampling, la más robusta entre las técnicas donde se establece un valor K que significa un porcentaje aleatorio de datos para el conjunto de testing, y una computadora portátil personal con procesador Intel Core i5 de 7th Gen sin la optimización que brinda Intel para aprovechar al máximo el rendimiento de la CPU y 8GB de RAM.

5.2 Resultados

Para comparar los tres modelos anteriormente descriptos, se utilizaron cuatro métricas de desempeño: BLEU, curva de aprendizaje, tiempo de entrenamiento y precisión.

5.2.1 BLEU

La métrica BLEU permite en el campo de la generación del lenguaje natural calificar una respuesta de salida a partir de las coincidencias con las respuestas candidatas para medir la precisión de las respuestas generadas.

Las Tablas 1, 2 y 3 describen la comparación de los resultados en la generación de las respuestas de salida de los modelos.

Epoch	BLEU		
	Modelo A	Modelo B	Modelo C
1000	0,29	0,05	0,89
3000	0,32	0,79	1,0
5000	0,97	1,0	0,97

Tabla 1. 400 unidades en cada capa

Epoch	BLEU		
	Modelo A	Modelo B	Modelo C
1000	0,30	0,38	0,95
3000	1,0	0,49	0,97
5000	0,79	0,72	0,98

Tabla 2. 700 unidades en cada capa

Epoch	BLEU		
	Modelo A	Modelo B	Modelo C
1000	0,35	0,25	0,99
3000	0,99	0,32	0,97
5000	0,72	0,99	0,99

Tabla 3. 1000 unidades en cada capa

Se puede observar que al aumentar la cantidad de epoch y la cantidad de unidades en cada capa de la red neuronal artificial se obtiene mejores resultados, sin embargo, esto conlleva una desventaja que es el tiempo de entrenamiento. El chatbot A.V.I. y el modelo B entre las 400 y 700 unidades y entre los 3000 y 5000 epoch muestran un buen desempeño pero el modelo C en 400 unidades tiene excelentes resultados.

5.2.2 Curva de Aprendizaje

La curva de aprendizaje describe el grado de éxito obtenido durante el aprendizaje en el transcurso del tiempo.

Las Figuras 3, 4 y 5 muestran la comparación de la curva de aprendizaje entre los distintos modelos.

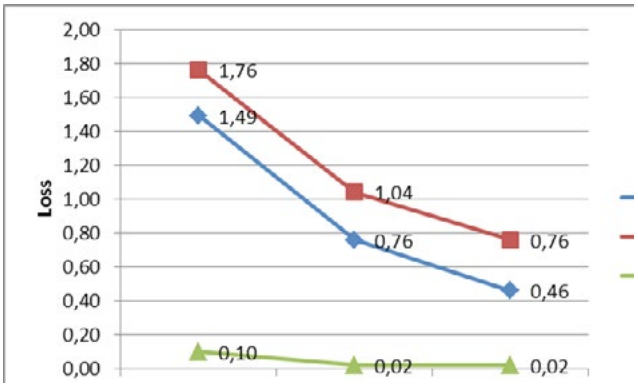

Figura 3. 400 unidades en cada capa

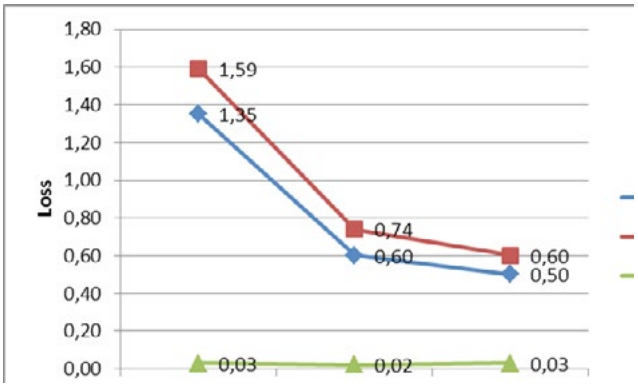

Figura 4. 700 unidades en cada capa

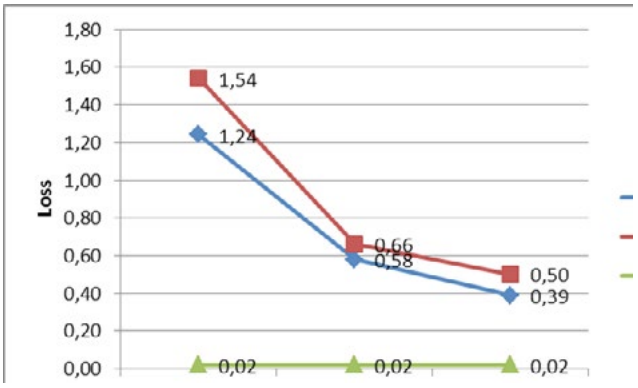

Figura 5. 1000 unidades en cada capa

Se puede observar que el modelo B tiene una baja curva de aprendizaje. El chatbot A.V.I. y el modelo C tienen una curva de aprendizaje deseable pero el modelo C es superior.

Tiempo de Entrenamiento

El tiempo de entrenamiento es un parámetro importante durante el diseño de la red neuronal artificial y depende de la cantidad de unidades en la capa, la cantidad de epoch y la cantidad de capas de la red.

Las Figuras 6, 7 y 8 muestran la comparación de los tiempos de entrenamiento entre los distintos modelos.

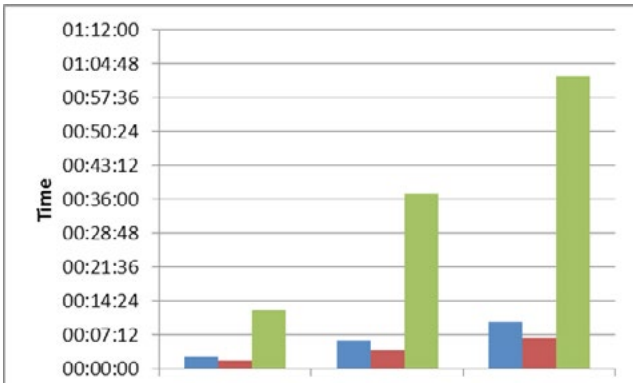

Figura 6. 400 unidades en cada capa

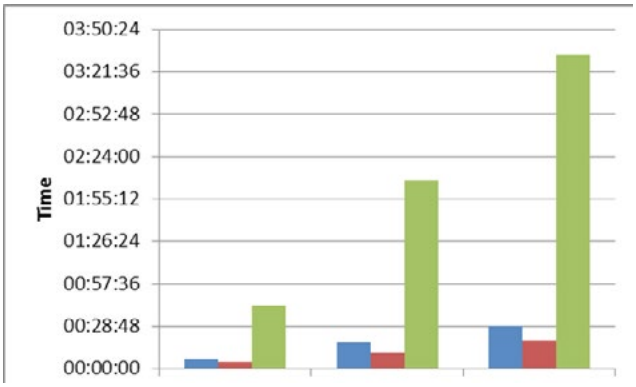

Figura 7. 700 unidades en cada capa

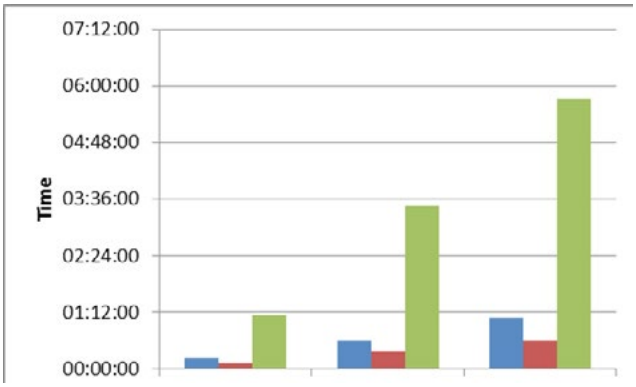

Figura 8. 1000 unidades en cada capa

Se puede observar que el modelo C requiere mayor tiempo de entrenamiento. En cambio, el chatbot A.V.I. y el modelo B muestran buenos resultados en 400 unidades y entre 1000 y 3000 epoch.

5.2.4 Precisión del Modelo

La precisión del modelos se utiliza para medir el rendimiento del algoritmo de forma interpretable.

Las Figuras 9, 10 y 11 muestran la comparación de la precisión del algoritmo entre los distintos modelos.

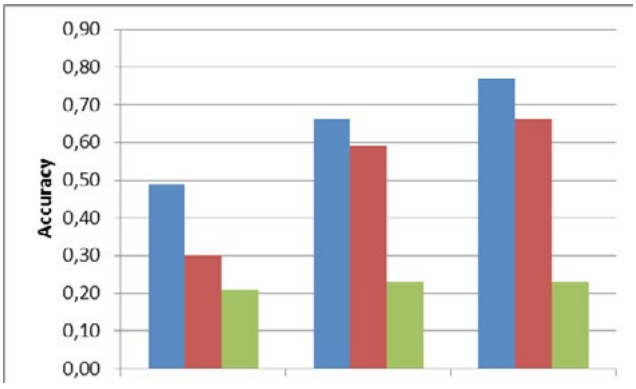

Figura 9. 400 unidades en cada capa

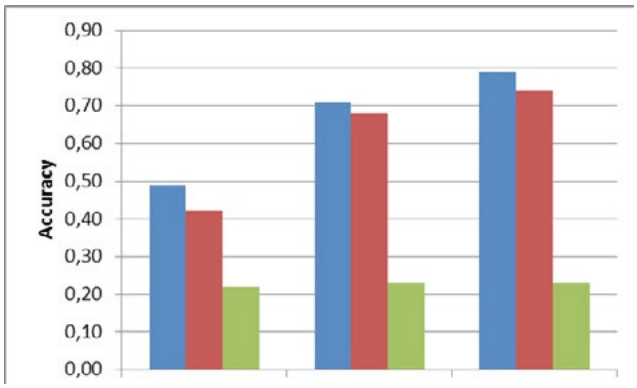

Figura 10. 700 unidades en cada capa

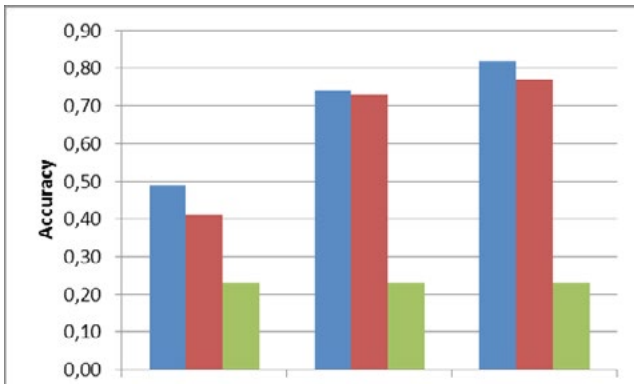

Figura 11. 1000 unidades en cada capa

Se puede observar que el modelo C tiene problemas con la tarea de clasificar. En cambio, el chatbot A.V.I. y el modelo B muestran buenos resultados entre las 400 y 700 unidades y entre los 3000 y 5000 epoch.

El experimento refleja como el chatbot A.V.I. brinda una contribución en la mejora de los resultados, por lo tanto, el desarrollo de un chatbot mediante la combinación de los dos modelos de conversación y la arquitectura utilizada en la red neuronal artificial podría ser la más preferible.

6. Conclusión y Trabajos Futuros

En este trabajo, para resolver la problemática que tienen los chatbots actuales dedicados a la atención al cliente sobre su carencia de flexibilidad para desarrollar diálogos reales debido a su dificultad para comprender el estilo de escritura de los usuarios, se presentó una arquitectura híbrida combinando los modelos de conversación basados en recuperación y generación con la capacidad de generar respuestas naturales y consistentes permitiendo mejorar la atención a los usuarios.

La evaluación de la arquitectura de la red neuronal artificial propuesta contra otros modelos ha mostrado resultados alentadores a través de las métricas de desempeño e inspira para realizar cambios significativos.

Como trabajo a futuro se modificará el módulo de recuperación para que el chatbot sea generalizable en idioma, se seguirá explorando otros algoritmos para mejorar los resultados obtenidos hasta ahora y se utilizará el Test de Turing porque dicho examen se utiliza en el campo de la inteligencia artificial para medir la capacidad que posee una máquina en exhibir un comportamiento inteligente similar al de un ser humano. A pesar que el modelo C tiene una codificación compleja y su tiempo de entrenamiento es superior a los otros dos modelos se lo volverá a investigar porque en (Serban, et al., 2017) el autor demuestra obtener buenos resultados con dicho modelo.

Referencias

Bhagwat, V. A. (2018). Deep Learning for Chatbots.

Csáky, R. K. (2017). Deep Learning Based Chatbot Models.

Ferraro, C & Restrepo, M. (2019). The Customer Service Experience: Aligning Channels with Evolving Consumer Expectations, Experiences & Behaviour.

Le, D. T, Nguyen, C. & Nguyen, K. (2018). Dave the debater: a retrieval-based and generative argumentative dialogue agent.

Nuez Ezquerra, A. (2018). Implemeting ChatBots using Neural Machine Translation techniques.

Peters, F. (2017-2018). Design and implementation of a chatbot in the context of customer support.

Reddy, T. (2017). How chatbots can help reduce customer service costs by 30%.

Serban, I. V, Sankar, C, Germain, M, Zhang, S, Lin, Z, Subramanian, S, Kim, T, Pieper, M, Chandar, S, Rosemary Ke, N, Rajeshwar, S, de Brebisson, A,

Sotelo, J. M. R, Suhubdy, D, Michalski, V, Nguyen, A, Pineau, J. & Benglo, Y. (2017). A Deep Reinforcement Learning Chatbot.

Slater, M. (2018). Twitter and chatbots help brands deliver top-notch customer service in Canada.

Song, Y, Yan, R, Li, X, Zhao, D. & Zhang, M. (2016). Two are Better than One: An Ensemble of Retrieval- and Generation-Based Dialog Systems.

Tiha, A. (2018). Intelligent Chatbot using Deep Learning.

Villar, V. B, Rodríguez, G & Rocha, F. G. (2018). Chatbots: A Systematic Mapping Study.

Villar, V. B, Rodríguez, G & Rocha, F. G. (2018). Interaction with Intelligent Conversation Agents: A case study.

Xu, A, Liu, Z, Guo, Y, Sinha, V. & Akkiraju, R. (2017). A New Chatbot for Customer Service on Social Media.

Yang, L, Hu, J, Qiu, M, Qu, C, Gao, J, Croft, W. B, Liu, X, Shen, Y. & Liu, J. (2019). A Hybrid Retrieval-Generation Neural Conversation Model.