

# Clasificación automática de preguntas en español en el dominio de comercio electrónico usando una red neuronal convolucional

Melissa A. De-León-Barrón<sup>1</sup>, Ana B. Rios-Alvarado<sup>1</sup>,  
Tania Y. Guerrero-Melendez<sup>1</sup>, Heidy Marisol Marín-Castro<sup>2</sup>,  
Jose L. Martinez-Rodriguez<sup>3</sup>

<sup>1</sup> Universidad Autónoma de Tamaulipas,  
Facultad de Ingeniería y Ciencias,  
México

<sup>2</sup> Universidad Autónoma de Tamaulipas,  
Consejo Nacional de Ciencia y Tecnología,  
México

<sup>3</sup> Universidad Autónoma de Tamaulipas,  
Unidad Académica Multidisciplinaria Reynosa,  
México

{melissa.barron, arios, tyguerre}@docentes.uat.edu.mx,  
{hmarisol, lazaro.martinez}@uat.edu.mx

**Resumen.** El comercio electrónico representa el proceso de comprar y vender productos y servicios por Internet. Uno de los principales retos que enfrenta el comercio electrónico es mejorar la experiencia del cliente, principalmente en el proceso de atención al cliente al momento de resolver preguntas de forma automática sobre los productos en venta o el proceso de compra. En este artículo se propone un modelo para la representación y clasificación de preguntas en español en el dominio de comercio electrónico. Se describe el diseño y desarrollo de un modelo de Aprendizaje Profundo basado en Redes Neuronales Convolucionales (CNN). La estrategia se centra en recuperar y asociar preguntas a una de las dos posibles clases: técnica (relacionados al producto) u operativa (relacionadas al proceso de compra). El conjunto de datos se representa mediante word embeddings utilizando las arquitecturas de Word2vec. Los resultados obtenidos demuestran que mediante la implementación de una red CNN se pueden desarrollar modelos que obtengan un buen desempeño para la tarea clasificación de preguntas en español.

**Palabras clave:** Minería de texto, aprendizaje profundo, CNN, word embedding, procesamiento del lenguaje natural.

## Automatic Classification of Spanish Questions in the E-Commerce Domain Using a Convolutional Neural Network

**Abstract.** E-commerce represents the process of buying and selling products and services over the Internet. One of the main challenges faced by e-commerce is to improve the customer experience, mainly in the customer service process when resolving questions automatically about the products for sale or the purchase process. This paper proposes a model for representing and classifying Spanish queries in the e-commerce domain. We describe the design and development of a Deep Learning model based on Convolutional Neural Networks (CNN). The strategy focuses on retrieving and associating questions to one of two possible classes: technical (related to the product) or operational (related to the purchasing process). The dataset is represented by word embeddings using Word2vec architectures. The results demonstrate that implementing a CNN network makes it possible to develop models that perform competently for the Spanish question classification task.

**Keywords:** Text mining, deep learning, CNN, word embedding, natural language processing.

### 1. Introducción

El surgimiento del Internet ha contribuido a hacer negocios y a mejorar el estilo de vida de las personas. Además, es uno de los principales requisitos para la existencia del comercio electrónico. El comercio electrónico es un concepto emergente que describe el proceso de comprar y vender productos, servicios e información vía Internet [3]. De acuerdo con un estudio realizado por Search Logistics [16], 2.140 billones de personas en todo el mundo compraron productos o servicios en línea en el 2021, eso es aproximadamente el 27 % de la población mundial.

Esto representa un aumento de 1.66 billones con respecto al 2016. Específicamente, en México durante el 2022 más de 63 millones de mexicanos adquirieron productos o servicios en Internet, lo que significa que nueve de cada 10 usuarios mayores de 18 años compró en línea [2]. En el transcurso del año 2020, una gran cantidad de países en todo el mundo atravesó por diversos tipos de medidas de restricción debido a la pandemia por el COVID-19, lo cual provocó que los consumidores y los negocios cambiaran drásticamente su estrategia de negocio.

Por esto, el Internet se convirtió en el principal medio de compra - venta, y de acuerdo a un estudio realizado por Think with Google reportó que la tendencia por realizar consultas sobre como realizar compras en línea creció en un 200 % en todo el mundo [1].

En la actualidad, existen una gran cantidad de plataformas de comercio electrónico operando todos los días alrededor del mundo, en donde una forma de comunicación entre los consumidores y los vendedores es a través de preguntas que los clientes potenciales realizan sobre los productos o servicios que ofrece dicha empresa.

Uno de los retos más importantes que se presentan en el comercio electrónico es la capacidad de los sistemas para responder automáticamente dichas preguntas.

Dentro de los sistemas de generación de respuestas automáticas existe un módulo de clasificación de preguntas. Es deseable que ante las preguntas en idioma español se pueda contar con modelos de representación y clasificación de textos cortos en ese idioma para apoyar la generación de respuestas automáticas.

Distintos proyectos han propuesto soluciones para la clasificación de textos cortos en varios contextos e idioma inglés, por ejemplo existen modelos para la clasificación de tuits [15], minería de opinión [17], sistemas de pregunta-respuesta [9], chatbots [4], entre otros. Sin embargo, son todavía pocos los modelos aplicados al español, debido a los retos que implica este lenguaje.

Por lo anterior, en este artículo se propone una estrategia para el diseño y desarrollo de un modelo de clasificación automática de preguntas en español para un dominio de comercio electrónico. El modelo está basado en una arquitectura de Red Neuronal Convolutiva (CNN) que utiliza las dos arquitecturas de Word2vec: Continuous Bag of words (CBOW) y Skip-gram para la representación del conjunto de datos. Para este estudio se evalúan diferentes configuraciones de la red CNN con el objetivo de encontrar la configuración óptima para alcanzar una mayor exactitud en la tarea de clasificación.

## **2. Trabajos relacionados**

Con el crecimiento de las redes sociales, el comercio electrónico y la comunicación en línea, la clasificación de textos cortos se ha convertido en un tema de tendencia en años recientes [13], un texto corto se puede encontrar en diferentes contextos o idiomas como por ejemplo tuits [10], mensajes de chat [?], reseñas de productos [18] o preguntas de clientes [11]. Una de las principales características de los textos cortos es su longitud, los mensajes de texto tienen alrededor de 70 caracteres, los títulos de noticias contienen 30 caracteres y los tuits no cuentan con más de 280 caracteres [19].

Por otro lado, la tarea de responder preguntas (en inglés, Question Answering) es una tarea importante en el Procesamiento del Lenguaje Natural que consiste en brindar respuestas en lenguaje natural a preguntas. Por esto, un elemento esencial de los sistemas de preguntas-respuesta es la clasificación de preguntas, la cual es una tarea que consiste en asignar una clase a una pregunta.

Por ejemplo, si un sistema de pregunta-respuesta conoce que el nombre de una persona es un tipo de respuesta a la pregunta “¿Quién es el presidente de México?” la respuesta se restringirá a una clase de tipo nombre propio. La clasificación de preguntas está basada en la clasificación de texto debido a que son similares en algunos aspectos pero difieren claramente en que una pregunta es usualmente más corta [20].

Kulkarni et al. [9], presentan un estudio para la respuesta de preguntas de clientes en páginas de productos de un dominio de comercio electrónico. Uno de los módulos de este sistema es la tarea de clasificación de preguntas. Ellos proponen implementar una red CNN (Convolutional Neural Network) y una representación de n-gramas. El sistema propuesto obtiene un resultado 66 % de precisión. En Kim [7] se presenta una serie de experimentos con redes convolucionales (CNN) y vectores pre entrenados de Word2vec por Mikolov et al. [12] con 100 billones de palabras de Google News.



**Fig. 1.** Diseño del modelo de clasificación de preguntas.

En este trabajo se muestra la arquitectura de la red CNN para el desarrollo de un modelo de clasificación de texto, dicha red neuronal consta de una arquitectura simple, ya que cuenta con una sola capa convolucional. El modelo reporta resultados de un 45 % de exactitud. Finalmente, en Kumar et al. [10] presentan un modelo de textos cortos basado en un enfoque de aprendizaje profundo. El conjunto de datos con el que se entrena dicho modelo está compuesto por reseñas de películas.

El modelo resultante reporta resultados de 99.07 % de exactitud para el entrenamiento y un 82.19 % de exactitud en la predicción. Con base en estos trabajos se observa que el aprendizaje profundo es efectivo para las tareas de clasificación de texto aunque para alcanzar dichos valores de exactitud se propone utilizar modelos pre entrenados de Word2vec.

### 3. Método propuesto

En esta sección se describe la metodología propuesta para la construcción de un modelo de clasificación de preguntas en español de un dominio de comercio electrónico. En la Figura 1 se muestra un diagrama con la metodología propuesta que consta de cuatro principales etapas: el preprocesamiento y representación de texto, el entrenamiento y configuración del modelo de clasificación, la evaluación del modelo se considera en el método propuesto.

#### 3.1. Conjunto de preguntas

En esta tarea se tiene el objetivo de recolectar y construir el conjunto de preguntas en español. Las preguntas se extrajeron del sitio de comercio electrónico: Mercado Libre México<sup>4</sup>, la cual fue seleccionada debido a que alberga una de las más relevantes plataformas de comercio en línea en todo América Latina y su crecimiento en el 2020 ha sido exponencial debido a la pandemia por COVID-19.

Por lo tanto, se pueden encontrar muchas preguntas realizadas por los clientes sobre las características de los productos, métodos de pago, devoluciones y reembolsos, servicio de envío, etc. El conjunto de preguntas se llevó a cabo mediante dos tareas:

- Recolección: Para descargar las preguntas de Mercado Libre México, se desarrolló un script de Python para realizar la extracción del contenido de la URL (Uniform Resource Locator) de diferentes productos contenidos en esta plataforma. Las preguntas extraídas fueron almacenadas en un archivo de texto.

<sup>4</sup> [www.mercadolibre.com.mx/](http://www.mercadolibre.com.mx/)

**Tabla 1.** Ejemplo de estructura del conjunto de preguntas.

Pregunta	Clase
¿En cuántos días hacen el envío a Acapulco?	1
¿Qué colores tienes disponibles?	2
¿Hay disponible en XS?	1
¿Compatible con Huawei Mate 20	2
...	...

- Etiquetado: El conjunto de datos consta de 500 preguntas divididas en dos clases (250 de cada una): operativas y técnicas. Las preguntas operativas son sobre métodos de pago, envíos, devoluciones y reembolsos. Por otro lado, las preguntas técnicas se refieren a las características de los productos, por ejemplo, tamaño o color. El conjunto de preguntas contiene dos atributos: la pregunta y la etiqueta. Es importante mencionar que un experto en lingüística etiquetó manualmente el conjunto de datos. La Tabla 1 muestra un ejemplo de la estructura del conjunto de preguntas.

### 3.2. Preprocesamiento de texto

El preprocesamiento es una tarea crucial que permite la limpieza y preparación del texto para posteriormente llevar a cabo su clasificación. Los textos en línea generalmente contienen mucho ruido por lo que contar con datos debidamente procesados permite reducir el ruido en el texto y mejorar el rendimiento del clasificador [5].

Las preguntas del conjunto de datos poseen una longitud de hasta 485 caracteres, aunque se pueden encontrar preguntas con una longitud mínima de 15 caracteres. Estas preguntas pueden llegar a estar conformadas hasta por 90 términos, que correspondería a las preguntas más extensas.

Por otro lado, algunas preguntas del conjunto cuentan con solo dos términos, debido a las características de las preguntas, este estudio se aborda como una tarea de clasificación de textos cortos. Sin embargo, para las preguntas en español, se realizó una estrategia de preprocesamiento específica compuesta por los siguientes pasos:

**1. Normalización.** Esta tarea es necesaria para procesar texto, específicamente, textos informales. Esto debido a que el lenguaje utilizado en las plataformas de comercio electrónico difiere de los textos escritos formalmente, por lo que se puede encontrar un uso del idioma diferente. En esta tarea se aplicaron algunas técnicas de limpieza de texto:

- Convertir el texto a minúsculas.
- Segmentar el texto en palabras (tokens).
- Eliminar acentos.
- Eliminar signos de puntuación.
- Eliminar caracteres numéricos.
- Eliminar caracteres especiales. En el español también se consideran como caracteres especiales la "ñ", y "ü", estas se reemplazan por las letras "n" y "u" respectivamente.

- Eliminar caracteres repetidos. Los usuarios usualmente repiten un carácter en una palabra para enfatizar y exagerar al momento de describir algo. Solamente se exceptúan los casos “ll” y “rr”, que son caracteres consecutivos válidos en el idioma español.
- Normalizar URL’s: una de las características que más se encuentran en las preguntas del conjunto de datos son las direcciones web, ya que los usuarios utilizan estos recursos para realizar una pregunta aún más específica. Una URL se considera una característica y se representa mediante la palabra “url”.

**2. Eliminación de palabras vacías.** Esta tarea permite eliminar palabras que aparecen con mucha frecuencia en las preguntas pero no son relevantes. En muchos idiomas, como el español, existe un conjunto de las palabras más comunes, como por ejemplo, los determinantes o las conjunciones (p.ej., el o y).

Este proceso permite reducir las palabras y mejorar el rendimiento del procesamiento del conjunto de datos. Para esta tarea se utilizó una lista de palabras vacías en español incluidas en el paquete de NLTK de Python<sup>5</sup>.

**3. Stemming.** El objetivo principal de esta técnica es eliminar cualquier sufijo y prefijo de las palabras con el fin de obtener la palabra raíz. El “stem” se obtiene después de aplicar un conjunto de reglas sin preocuparse por la parte del discurso (POS) o el contexto de ocurrencia de la palabra [6].

En este estudio se utilizó el Snowball Stemmer para el idioma español implementado en el paquete de NLTK de Python. El texto preprocesado se guarda en un archivo JSON (Java Script Object Notation) que se utiliza para la representación de texto de las preguntas.

### 3.3. Representación de texto

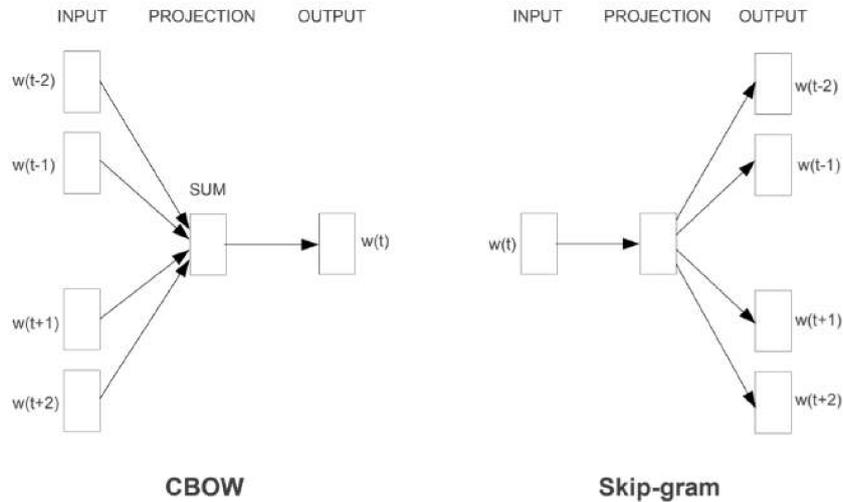
En la etapa de representación de texto se transforma el texto preprocesado de la pregunta a una representación matemáticamente computable, esta representación en la mayoría de los casos es un vector ponderado. Para llevar a cabo la representación de las preguntas para su posterior clasificación con una red CNN se utilizó el modelo Word2vec.

Word2vec es un método que permite obtener word embeddings para una mejor representación de las palabras. Su principal objetivo es transformar un espacio de características de gran dimensionalidad a vectores de características de baja dimensión al preservar la similitud contextual del conjunto de datos. A continuación se describen los pasos para la construcción de un modelo de word embeddings utilizando las arquitecturas de Word2vec.

- Construcción de Modelo Word2vec. Como primer paso se debe construir un modelo Word2vec, para esto se utilizó la biblioteca Gensim<sup>6</sup> de Python, ya que provee de una clase para trabajar con modelos Word2vec permitiendo utilizar las dos arquitecturas de Word2vec: Continuous Bag Of Words (CBOW) y Skip-gram. Para que Word2vec genere los vectores se necesita que se proporcione como entrada los textos de las preguntas.

<sup>5</sup> [pypi.org/project/nltk/](https://pypi.org/project/nltk/)

<sup>6</sup> [pypi.org/project/gensim/](https://pypi.org/project/gensim/)



**Fig. 2.** Arquitecturas de Word2vec: CBOW predice la palabra actual basado en el contexto. Skip-gram predice las palabras contexto dada una palabra (basada en Mikolov et.al [12]).

- Representación de las preguntas. Para representar las preguntas se realiza la transformación del texto a una secuencia de números, utilizando la clase Tokenizer de Keras<sup>7</sup>, mediante la implementación de un método se vectoriza el conjunto de preguntas, convirtiendo cada uno de los textos en una secuencia de enteros. Un punto importante es que los vectores de las preguntas codificadas se deben normalizar a una longitud máxima, esta con respecto a la longitud máxima de los textos que conforman el conjunto de entrenamiento y se rellenan con ceros las dimensiones sobrantes.
- Construcción de la matriz de embeddings. La matriz de embeddings tiene el objetivo de alimentar la capa de embedding de una red neuronal de aprendizaje profundo. Esta capa de la red neuronal se utiliza principalmente en aplicaciones relacionadas con el procesamiento del lenguaje natural, como el modelado del lenguaje.

Una matriz de word embeddings es una lista de todas las palabras del vocabulario y sus respectivos embeddings o pesos. La dimensión que se definió para la construcción del modelo Word2vec para este estudio fue de 300 dimensiones.

### 3.4. Configuración del modelo word2vec

Para la implementación del modelo de Word2vec se utilizan las dos arquitecturas propuestas por el modelo, las cuáles utilizan una red neuronal de tres capas (1 capa de entrada, 1 capa oculta, 1 capa de salida): CBOW y Skip-gram, así como se muestra en la Figura 2. Para que Word2vec genere los vectores se necesita que se proporcione como entrada los textos de las preguntas, a continuación se especifican los parámetros que se configuraron para construir dicho modelo.

<sup>7</sup> keras.io/

- Sentences: son los datos de entrada, en este caso son los textos de las preguntas. Como entrada se tiene un total de 2,000 preguntas, esto debido a que se pretende que el modelo se entrene con un mayor número de textos para que modele un vocabulario más amplio.
- Workers: 2. Número de hilos que se corren en paralelo.
- Size: 300. Normalmente, el tamaño que se utiliza es de 300 dimensiones, ya que se obtienen vectores más densos y más ricos con respecto a la relevancia de cada palabra en cada dimensión del vector [14].
- Window: 2. Es el número de palabras que se encuentran antes o después de cada palabra objetivo (word target) con el fin de tomarlas en cuenta y entrenar el modelo.
- Min\_count: 1. Es el mínimo de ocurrencias de una palabra para ser tomada en cuenta, ya que se define con un valor de 1, todas las palabras se incluyen en el modelo.
- Sg: 0 y 1. Es la arquitectura a utilizar; 1:Skip-gram y 0: CBOW. Se configuran ambas opciones con el fin de realizar experimentos con las dos arquitecturas de Word2vec.
- Iter: 40. Número de iteraciones (épocas) de entrenamiento sobre el conjunto de datos.

### 3.5. Configuración de red neuronal convolucional (CNN)

Las Redes Neuronales Convolucionales (en inglés, Convolutional Neural Networks) son arquitecturas de redes neuronales que cuentan con diversas capas ocultas, lo que provoca que cuente con altos niveles de profundidad. Típicamente, en una CNN,  $W_j$  es una convolución y  $p$  es un rectificador  $\max(x, 0)$  o una función sigmoide.

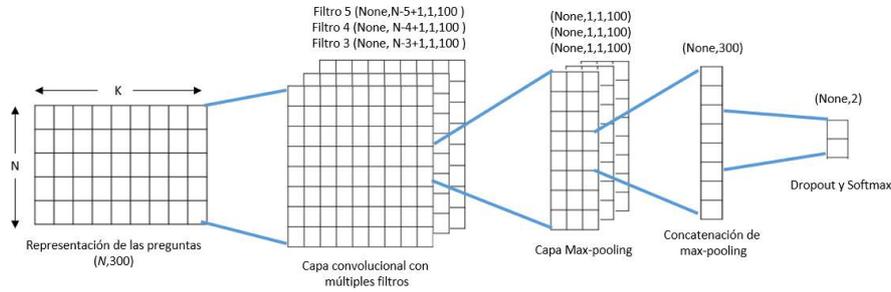
En resumen, se puede ver a  $W_j$  como una pila de filtros convolucionales. De este modo, las capas son mapas de filtro y cada capa se puede formalizar como una suma de convoluciones de la capa anterior, tal y como se define en la Ecuación 1 [8]:

$$x_j(u, k_j) = p\left(\sum_k (x_{j-1}(\cdot, k) \cdot W_j, k_j(\cdot, k))(u)\right). \quad (1)$$

Para la clasificación de preguntas se utiliza una arquitectura basada en la propuesta por Kim [7]. Esta consta de una capa embedding (representación de palabras por vector), una capa convolucional, una capa max pooling, una capa dropout y una capa de salida fully-connected. En la Figura 3 se muestra la estructura general de un modelo de clasificación de preguntas con una red neuronal convolucional.

Para este estudio se implementó una red CNN para la clasificación de preguntas en español de un dominio de comercio electrónico. A continuación se describen las características del modelo:

- Capa Embedding: La dimensión del vector de entrada se define a la máxima longitud detectada en los textos de las preguntas  $N$ . Por lo que, el vector resultante se define con una dimensión de (55,300), dicho vector es la entrada a la red convolucional. La matriz de word embedding que se obtuvo mediante Word2vec se carga en esta capa con el fin de realizar un tipo de aprendizaje de transferencia a través del modelo pre-entrenado de word embeddings.



**Fig. 3.** Arquitectura de una red neuronal convolucional para la clasificación de preguntas. Basada en Kim [7].

- **Capa Convolutiva:** Se utilizan tres filtros diferentes con tamaños de kernel  $k = [3, 4, 5]$  de un ancho de 100. Además, la función ReLU (Rectified Linear Unit) es utilizada como una función de activación no lineal para calcular el mapa de características de la capa de convolución.
- **Capa Pooling:** Para este problema se opta por la función max-pooling, debido a que es ampliamente adoptada para este tipo de tareas. Posteriormente, los valores máximos de todos los mapas que se hayan creado se concatenan para dar como resultado el vector de la pregunta.
- **Capa Dropout:** Esta capa mantiene activa solo una neurona durante el entrenamiento, para esto se debe determinar una probabilidad  $p=0.5$ .
- **Capa Softmax:** La capa softmax fully-connected nos permite transformar los valores de salida de la clase positiva y negativa en probabilidades normalizadas utilizando la función softmax definida por la Ecuación 2:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{t=0}^1 e^{z_t}} \text{ para: } i = 0, 1. \quad (2)$$

## 4. Experimentos

Los experimentos tienen el objetivo de llevar a cabo el entrenamiento y evaluación de seis configuraciones de redes CNN. De acuerdo al escenario de evaluación definido se pueden obtener las métricas de evaluación para posteriormente realizar un análisis de los resultados obtenidos por los modelos de clasificación CNN.

### 4.1. Entrenamiento del modelo

En esta etapa se entrena el modelo propuesto y se utiliza el escenario de evaluación definido. En la Tabla 2 se muestran las seis configuraciones implementadas para este estudio.

- **Implementación:** La red neuronal CNN se implementó mediante los módulos de Keras<sup>8</sup> y Tensorflow<sup>9</sup> para Python 3.7.

<sup>8</sup> keras.io

<sup>9</sup> www.tensorflow.org/?hl=es-419

**Tabla 2.** Configuraciones de redes neuronales CNN para la clasificación de preguntas.

Conf.	Tamaño filtros	No. filtros	Optimizador	Factor de aprendizaje	Épocas	Tamaño Batch
CNN1-CB	3,4,5	300	adadelta	0.001	30	64
CNN2-CB	3,4,5	300	adadelta	0.001	25	50
CNN3-CB	3,4,5	300	adadelta	0.001	30	50
CNN4-SG	3,4,5	300	adadelta	0.001	30	64
CNN5-SG	3,4,5	300	adadelta	0.001	25	50
CNN6-SG	3,4,5	300	adadelta	0.001	30	50

- Configuración del modelo: Se configuraron diferentes modelos CNN con el fin de compararlos y seleccionar el modelo óptimo para la clasificación de preguntas en español. Se utilizaron las dos arquitecturas de Word2vec (CBOW y skip-gram) previamente entrenadas. Lo anterior, con el fin de conocer que arquitectura se adapta mejor a la tarea de clasificación de preguntas.

#### 4.2. Métricas

Los modelos de clasificación de preguntas obtenidos se evalúan utilizando las siguientes métricas: precisión, cobertura, medida F y exactitud. Estas métricas son calculadas para cada una de las clases mediante las siguientes formulas:

$$\text{exactitud} = \frac{\text{total TP} + \text{total TN}}{\text{total ejemplos}}, \quad (3)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (4)$$

$$\text{cobertura} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (5)$$

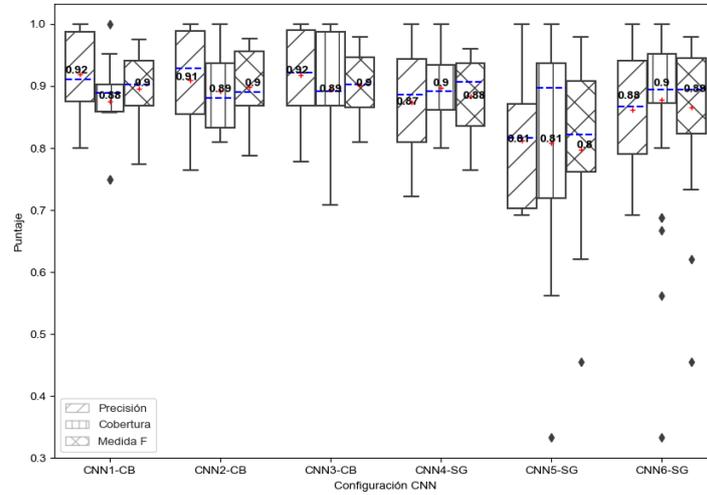
$$\text{medidaF} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \quad (6)$$

donde, TP denota el número de positivos reales, TN, los positivos negativos; FP, número de falsos positivos y FN denota los falsos negativos.

#### 4.3. Predicción del modelo

En esta etapa se utiliza la técnica de validación cruzada de “k=10” iteraciones. Por lo que, el conjunto de datos de las preguntas se divide en 10 grupos o subconjuntos. El primer grupo se selecciona para pruebas y los grupos restantes para el entrenamiento.

En cada iteración, se utiliza el 80 % de los datos para entrenamiento y el 20 % para pruebas. Los resultados de exactitud, precisión, cobertura y medida F de cada iteración fueron promediados.



**Fig. 4.** Resultados de las métricas de Precisión, Cobertura y Medida F de la validación cruzada 10 iteraciones de las configuraciones de la red CNN.

## 5. Resultados y discusión

En la Figura 4 se presenta un diagrama de cajas con los resultados obtenidos en la evaluación usando validación cruzada de 10 iteraciones, por cada una de las configuraciones de las redes CNN. Los modelos CNN1-CB y CNN3-CB obtuvieron resultados muy similares. El modelo CNN1-CB obtuvo un 92 % de precisión, 88 % de cobertura y un 90 % de medida F. Por otro lado, el modelo CNN3-CB obtuvo un 92 % de precisión, 89 % de cobertura y 90 % de medida F.

Estos dos modelos cuentan con una alta precisión y una alta cobertura, llegando a tener más de un 80 % en ambas métricas. En la Tabla 3 se muestran los resultados de exactitud y error obtenidos durante el entrenamiento por los seis modelos de clasificación. Con los resultados obtenidos se observa que los modelos CNN1-CB, CNN2-CB y CNN3-CB muestran un desempeño similar. El modelo CNN3-CB obtuvo un 91 % de exactitud y un error del 0.339, los modelos CNN1-CB y CNN2-CB obtuvieron un 90 % de exactitud con un error del 0.320 y 0.325 respectivamente.

Dado los resultados obtenidos con respecto a las métricas de exactitud, precisión, cobertura y medida F, se observó que los modelos que obtuvieron el mejor desempeño en la etapa de entrenamiento implementan un modelo Word2vec con arquitectura CBOW. Cabe mencionar que para que los modelos de clasificación basados en redes profundas puedan obtener mejores resultados se necesita de conjuntos de datos más grandes para construir un modelo Word2vec con una gran cantidad de palabras y por ende poder utilizar el modelo preentrenado en el entrenamiento de una red neuronal convolucional, con el fin de obtener un modelo más robusto.

Con los experimentos realizados se puede observar que para construir un modelo de clasificación que obtenga un alto nivel de precisión en la clasificación de preguntas en español se puede hacer uso de un enfoque de aprendizaje profundo y la arquitectura CBOW de Word2vec para obtener resultados por encima de un 80 % de precisión.

**Tabla 3.** Resultados de exactitud y error de los modelos CNN entrenados para la clasificación de preguntas.

Conf.	Exactitud	Error
CNN1-CB	0.90	0.320
CNN2-CB	0.90	0.325
CNN3-CB	0.91	0.339
CNN4-SG	0.89	0.374
CNN5-SG	0.82	0.484
CNN6-SG	0.90	0.371

## 6. Conclusiones

En este trabajo se propone un enfoque de aprendizaje profundo para el desarrollo de un modelo de clasificación de textos cortos en el idioma español. Dicho modelo permite representar y clasificar las preguntas de un dominio de comercio electrónico a una de las dos posibles clases: operativa o técnica.

La estrategia utilizada consiste en contar con un conjunto de datos etiquetado manualmente para posteriormente realizar un preprocesamiento y representación de los textos. Finalmente, se lleva a cabo el entrenamiento y evaluación de los modelos de clasificación basados en una Red Neuronal Convolutiva.

Los modelos de aprendizaje profundo, como Word2Vec se utilizan para obtener mejores representaciones vectoriales de palabras y mejorar la precisión de los clasificadores entrenados con algoritmos tradicionales de aprendizaje automático.

Con base en los experimentos realizados se puede observar que para aplicar un enfoque de aprendizaje profundo que permita obtener modelos de clasificación por encima del 90 % de exactitud se debe utilizar un conjunto de datos mucho más grande, que contenga millones de preguntas. Lo anterior, con el fin de entrenar un modelo de word embeddings más robusto, que sea capaz de representar un mayor número de palabras referentes al dominio de estudio.

Como trabajo futuro se pretende identificar y etiquetar el conjunto de datos con más clases con el fin de desarrollar un modelo de clasificación de preguntas multiclase. Además, implementar otras arquitecturas de redes neuronales de aprendizaje profundo como las Redes Neuronales Recurrentes (RNN) y las Redes Long Short Term Memory (LSTM), con el fin de comparar su desempeño con respecto a las redes CNN.

## Referencias

1. Blacksip México: Reporte de Industria: El e-commerce en México 2020. Blacksip México (2020)
2. Forbes: Comercio electrónico en México repuntó 23 % en 2022, revela estudio de la AMVO (2023) [www.forbes.com.mx/comercio-electronico-en-mexico-repunto-23-en-2022-revela-estudio-de-la-amvo/](http://www.forbes.com.mx/comercio-electronico-en-mexico-repunto-23-en-2022-revela-estudio-de-la-amvo/)
3. Gaffar-Khan, A.: Electronic commerce: A study on benefits and challenges in an emerging economy. *Global Journal of Management and Business Research: B Economics and Commerce*, vol. 16, no. 1, pp. 19–22 (2016)

4. Gupta, S., Borkar, D., Mello, C. D., Pati, S.: An E-Commerce website based chatbot. *International Journal of Computer Science and Information Technologies*, vol. 5, pp. 1483–1485 (2015)
5. Haddi, E., Liu, X., Shi, Y.: The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, vol. 17, pp. 26–32 (2013) doi: 10.1016/j.procs.2013.05.005
6. Jivani, A. G.: A comparative study of stemming algorithms. *International Journal of Computer Applications in Technology*, vol. 2, no. 6 (2011)
7. Kim, Y.: Convolutional neural networks for sentence classification (2014) doi: 10.48550/ARXIV.1408.5882
8. Koushik, J.: Understanding convolutional neural networks (2016) doi: 10.48550/ARXIV.1605.09081
9. Kulkarni, A., Mehta, K., Garg, S., Bansal, V., Rasiwasia, N., Sengamedu, S.: ProductQnA: Answering user questions on E-commerce product pages. pp. 354–360 (2014) doi: 10.48550/ARXIV.1408.3829.
10. Kumar, S., Zymbler, M.: A machine learning approach to analyze customer satisfaction from airline tweets. *Journal of Big Data*, vol. 6, no. 1 (2019) doi: 10.1186/s40537-019-0224-1
11. Li, Y., Miao, Q., Geng, J., Alt, C., Schwarzenberg, R., Hennig, L., Hu, C., Xu, F.: Question answering for technical customer support. *Natural Language Processing and Chinese Computing*, Springer International Publishing, vol. 11108, pp. 3–15 (2018) doi: 10.1007/978-3-319-99495-6\_1
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013) doi: 10.48550/ARXIV.1301.3781
13. MonkeyLearn: Short Text Classification. MonkeyLearn (2020) [monkeylearn.com/short-text-classification/](https://monkeylearn.com/short-text-classification/)
14. Parodi, G., Cantos-Gómez, P., Howe, C.: *Lingüística de corpus en español*. Routledge (2022)
15. Qasem, M., Thulasiram, R., Thulasiram, P.: Twitter sentiment classification using machine learning techniques for stock markets. In: *International Conference on Advances in Computing, Communications and Informatics*, pp. 834–840 (2015) doi: 10.1109/icacci.2015.7275714
16. Search Logistics: Ecommerce Statistics 2023. Search Logistics (2023) [www.searchlogistics.com/grow/statistics/ecommerce-statistics](https://www.searchlogistics.com/grow/statistics/ecommerce-statistics)
17. Sharma, R., Nigam, S., Jain, R.: Opinion mining of movie reviews at document level. *International Journal on Information Theory*, vol. 3, no. 3 (2014) doi: 10.48550/arXiv.1408.3829
18. Singla, Z., Randhawa, S., Jain, S.: Sentiment analysis of customer product reviews using machine learning. In: *International Conference on Intelligent Computing and Control* (2017) doi: 10.1109/i2c2.2017.8321910
19. Song, G., Ye, Y., Du, X., Huang, X., Bie, S.: Short text classification: A survey. *Journal of Multimedia*, vol. 9, no. 5 (2014) doi: 10.4304/jmm.9.5.635-643
20. Xin, L., Xuan-Jing, H., Li-de, W.: Question Classification using Multiple Classifiers. In: *Proceedings of the Fifth Workshop on Asian Language Resources and First Symposium on Asian Language Resources Network* (2005)