

Aplicación de análisis de sentimientos en Twitter mediante RNC para NLP con Python.

Proyecto Intermedio

Argenis Hernández García

Maestría en Ciencias de la Computación

Matemáticas para las Ciencias de la Computación

Profesor: Dr. Ricardo Barrón Fernández

Centro de Investigación en Computación, IPN

November 2, 2021

Contents

	Página
1 SECCIÓN	2
1.1 Justificación	2
1.2 Objetivos	2
1.3 Aportaciones	2
2 SECCIÓN	3
2.1 Estado del arte	3
3 SECCIÓN	4
3.1 Fundamentos	4
3.2 Solución Propuesta	4
4 SECCIÓN	6
4.1 Resultados	6
4.2 Conclusión	8
Bibliography	9

1 SECCIÓN

1.1 Justificación

El proyecto se plantea para indagar y profundizar en el análisis de sentimientos mediante una técnica particularmente interesante que son las redes neuronales convolucionales, con el interés de partir desde imágenes de tuits, desarrollar el proceso de conversión a texto y posteriormente poder aplicar técnicas de procesamiento de lenguaje natural. Ya que es un tema de intereses para el desarrollo de tesis de posgrado en el CIC. Buscando de esta forma las técnicas y métodos mas adecuados como futuro plan de ataque.

1.2 Objetivos

General: Desarrollar un analizador de sentimientos para clasificar Tuits como positivos o negativos mediante entrenamiento.

Específicos:

- Realizar la conversión de imágenes a texto.
- Realizar un Preprocesamiento de datos.
- Construir el modelo a base de capas de Convolución y Concatenar funciones lineales.
- Desarrollar una aplicación práctica, entrenando y testeando el modelo.

1.3 Aportaciones

Se busca demostrar como a partir de un conjunto de imágenes se puede aplicar el procesamiento de lenguaje natural haciendo una conversión a texto y entrenando una red neuronal por capas que pueda clasificar de forma correcta los sentimientos en frases ordinarias escritas en una red social muy popular la cual es Twitter. Demostrando las RNC pueden ser una herramienta potente para este tipo de análisis donde en el campo de la industria se puedan aplicar estas técnicas para analizar grandes cantidades de texto partiendo de conjuntos (datasets) de imágenes que son datos no estructurados. Demostrando porqué el NLP y la IA son campos de la Ciencias de datos.

2 SECCIÓN

2.1 Estado del arte

No.	Título	Autores	Publicación que emite	Año	Ideas principales
1	Análisis de sentimiento en los procesos de búsqueda de información en internet	Gemma García López	UNIVERSIDAD COMPLUTENSE DE MADRID (Tesis Doctoral).	2020	La presente investigación está centrada en el análisis del binomio lenguaje-emoción como elemento diferenciador en esa interacción humano-máquina, concretamente en los procesos de búsqueda de información en Internet. Aplicando el método empírico-analítico y utilizando diferentes herramientas de análisis de sentimiento, analizamos la presencia de emociones tanto en las consultas como en los resultados de búsqueda.
2	Aplicación de un modelo híbrido de aprendizaje profundo para el análisis de sentimientos en Twitter	Montañés, R.; Aznar, R.; Del Hoyo, R.	CEUR Workshop Proceedings 2172	2018	Este trabajo pretende explorar modelos presentes en el estado del arte actual del aprendizaje profundo aplicado al modelado y clasificación de texto. Se ha analizado el uso de modelos de redes convolucionales (CNN), Long short Term Memory (LSTM), LSTM bidireccionales (BI-LSTM) y una aproximación híbrida entre CNN y LSTM para su uso en el análisis de sentimiento en Twitter. Se ha optado por la combinación CNN- LSTM ya que integra los beneficios de ambos modelos.
3	Análisis de Sentimiento de Tweets en español utilizando SVM y CNN.	Aiala Rosa, Luis Chiruzzo, Mathias Etcheverry, Santiago Castro.	RETUYT en TASS	2017	En este artículo se presentan clasificadores basados en SVM y Redes Neuronales Convolucionales (CNN) para la competencia TASS 2017 de clasificación de sentimientos de tweets. El clasificador con mejor desempeño en general utiliza una combinación de SVM y CNN. El uso de Word embeddings resultó particularmente importante para mejorar el desempeño de los clasificadores.
4	Modelo basado en aprendizaje profundo para el análisis de sentimiento en español.	Bermejo Escobar, Danitza Yvette; Vizcarra Aguilar, Gerson Waldyr	Universidad nacional del altiplano (Tesis).	2019	La presente tesis propone tres modelos basados en aprendizaje profundo para abordar la tarea de análisis de sentimiento de tuits en español. El objetivo es mejorar los resultados obtenidos por métodos anteriores. Para ello, se ha realizado el preprocesamiento de los datos y la generación de representaciones de palabras que serán las entradas de los modelos. Seguidamente, se implementaron las redes neuronales recurrente, convolucional y un híbrido de ambos.
5	Análisis del Sentimiento Político en Twitter durante las Elecciones Congresales 2020 en el Perú	Alva-Segura, Daniel Abraham	Universidad Internacional de la Rioja. (Tesis de Maestría).	2021	El segundo objetivo es extraer los datos de Twitter de octubre 2019 a enero 2020 usando las palabras clave del nombre del partido político y que el tuit sea en español, utilizando técnicas del preprocesamiento de datos para limpiar los tuits. Se han creado y optimizado modelos como Naive Bayes, Máquina de vectores de soporte (SVM) y redes neuronales convolucionales (RNC) utilizando la metodología Cross Industry Standard Process for Data Mining (CRISP-DM) para analizar el sentimiento de dichos datos a nivel de documento.
6	Análisis de sentimiento con aprendizaje profundo en un entorno de Big Data	Civera Sancho, Javier	Universidad de Zaragoza, EINA	2019	En este trabajo se plantea el diseño e implementación de un sistema de análisis de sentimientos sobre textos en español, basado en tecnologías de aprendizaje profundo y big data. Estos algoritmos convenientemente escalados y configurados forman arquitecturas complejas de computación, que permiten llevar a cabo un procesamiento masivo y paralelo de información. Alimentados con los grandes volúmenes de datos disponibles abiertamente a través de internet, dan lugar al desarrollo del paradigma de computación conocido como \textit{Deep Learning} o aprendizaje profundo.
7	Análisis de sentimiento a nivel de documento en críticas de cine en español	Maritza Flores Domínguez, José Luis Tapia Fabela	Research in Computing Science 149(8)	2020	El método propuesto emplea la red neuronal back-propagation para clasificar el corpus en cinco clases. Para esto se crearon diferentes muestras de entrenamiento normalizadas con tres tipos de preprocesamiento y se representaron mediante un conjunto de vectores binarios. Los resultados obtenidos muestran que el accuracy obtenido de evaluar la red es similar al de los clasificadores del estado del arte, debido a la muestra de entrenamiento, la configuración de la red, el número de clases utilizadas, así como al número de críticas empleadas.

3 SECCIÓN

3.1 Fundamentos

Las Redes Neuronales Convolucionales son aplicadas en procesos de visión artificial, pero también muestran resultados extraordinarios para procesamiento de clasificación en texto. Siendo una de las principales tareas de las RNC, donde destacan por su flujo de funcionamiento. En una aplicación de NLP, el funcionamiento de la red puede obtener buenos resultados en la evaluación al tener un entrenamiento profundo por varias capas de una RNC.

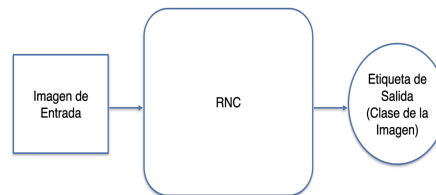


Figure 3.1: Esquema funcional del proceso de RNC.

3.2 Solución Propuesta

La conversión de una imagen a texto en representación matricial, de modo que cada palabra se transformará en un vector de coordenadas. Transformando palabras en vectores numéricos para implantarlos dentro de la red neuronal. De modo que que la red podrá interpretar el texto haciendo una relación entre palabras para proceder a clasificar de forma matemática una representación por dimensiones.

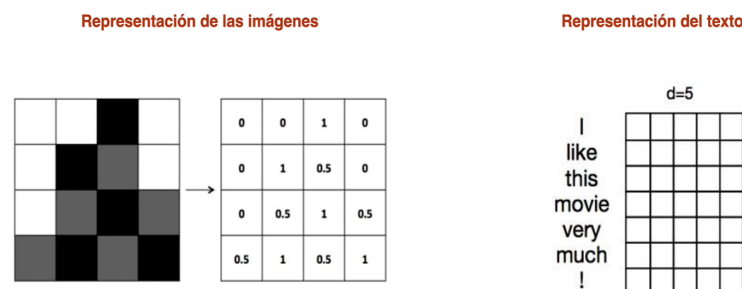


Figure 3.2: Conversión de representación de imagen a texto por vectores.

Por medio de representaciones vectoriales, podremos aplicar operaciones matemáticas a nuestra conversión de palabras, esto para obtener una relación entre cada palabra, de este modo podremos crear conocimiento en nuestra red neuronal. De modo que al ubicar en nuestro plano de dimensiones a cada vector, obtenemos que las posiciones están relacionadas al significado de cada palabra tal y como se representa en la siguiente imagen. Mediante el input de entrada y la multiplicación de la matriz

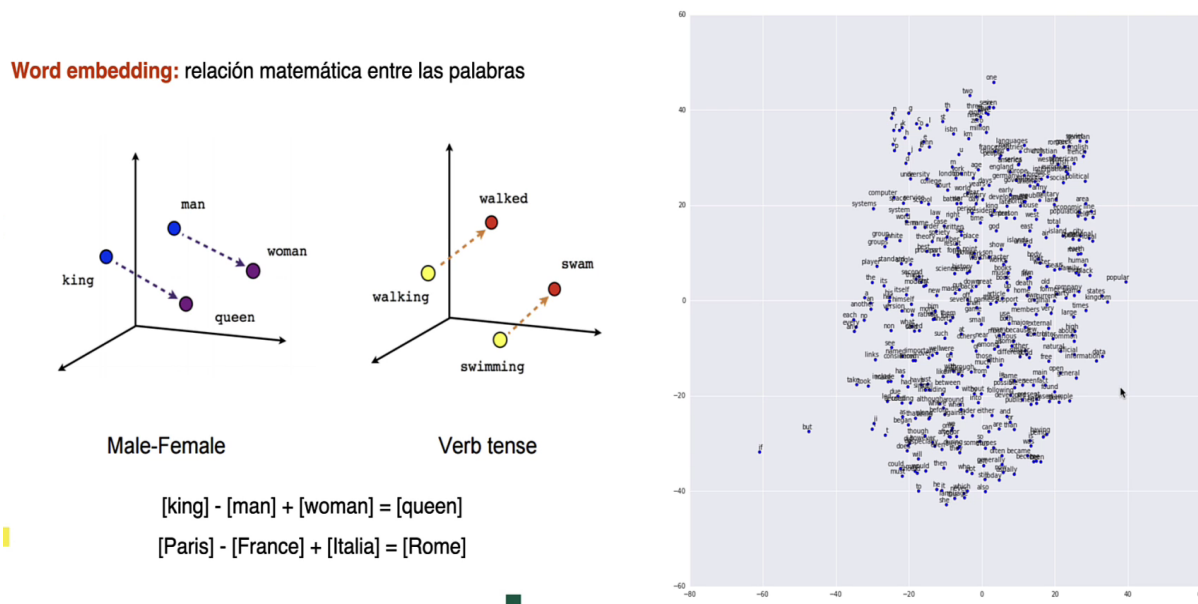


Figure 3.3: Representaciones de Word Embedding. Forma vectorial y matricial.

embebida obtenemos un Hidden para volverlo a multiplicar por el contexto de la matriz resultante mediante las capas de convolución para obtener una salida de un vector con una correlación semántica. De este modo sabremos que palabras salen cerca de la palabra codificada al inicio.

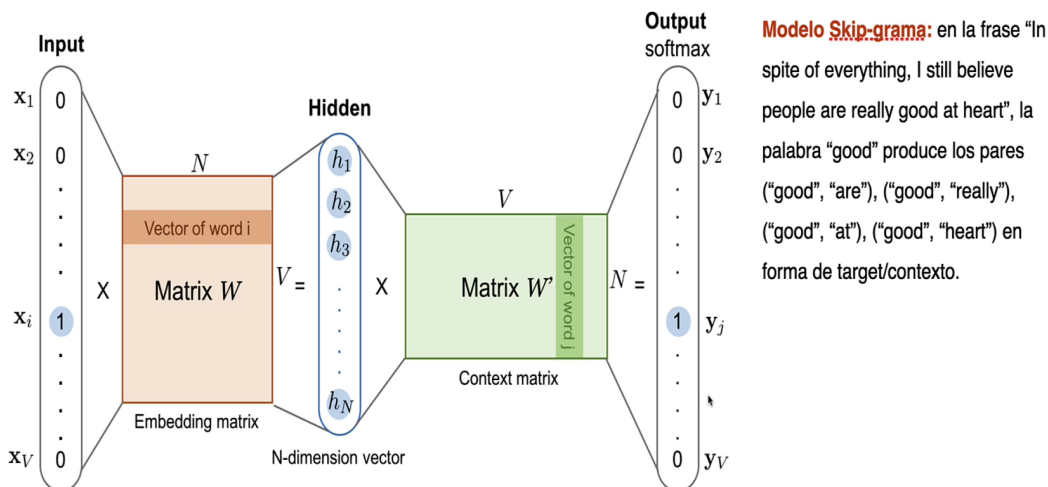


Figure 3.4: Modelo Skip-gram para representar el contexto de las palabras.

4 SECCIÓN

4.1 Resultados

Para proceder con los objetivos, se obtienen los resultados de la carga de los datasets ya en texto y mandamos a llamar una lista como ejemplo de la visualización de los campos que contienen los Tuits que servirán como entrenamiento en pasos adelante. Posterior a ello, se realiza la limpieza de los

	sentiment	id	date	query	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zi - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	matycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwidedeclan no, it's not behaving at all...

El conjunto de datos de testing tiene 3 etiquetas diferentes (una negativa, una positiva y una neutra), mientras que el conjunto de datos de entrenamiento tiene solo dos, por lo que no usaremos el archivo de testing y dividiremos el archivo de entrenamiento más tarde nosotros mismos.

Figure 4.1: Presentación de los datos cargados originales.

datos para dejar unicamente el identificador y el texto. Esto como parte de Pre-procesado de datos, al igual que se realiza la tokenización, se coloca un padding y dividimos en los conjuntos de entrenamiento y de testing.

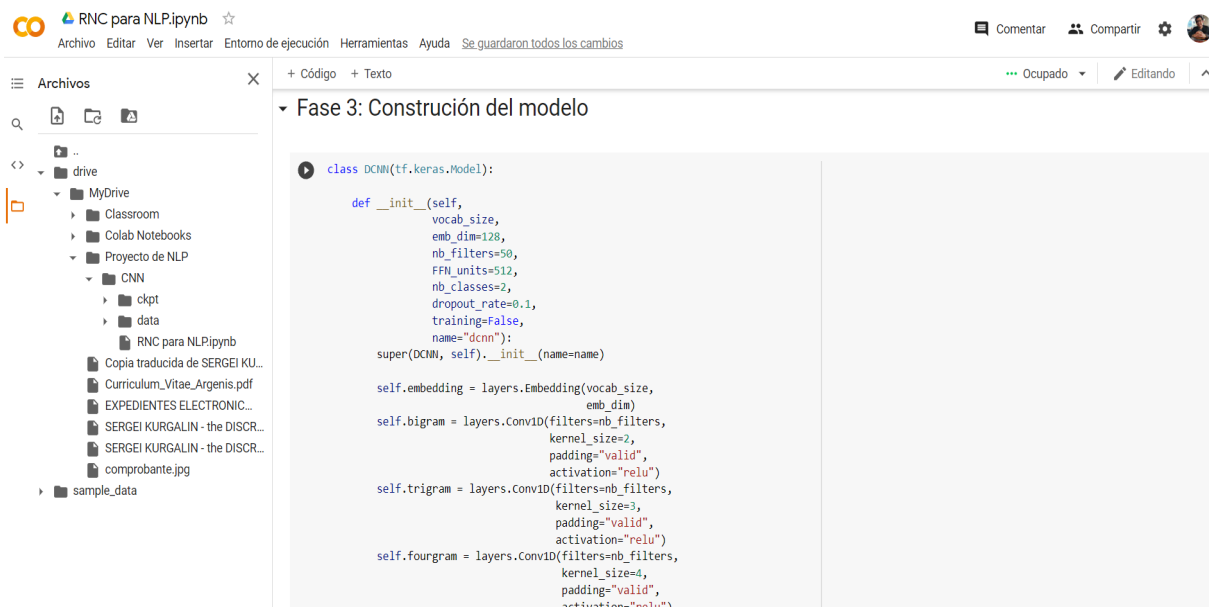
```
[ ] data_labels = data.sentiment.values
data_labels[data_labels == 4] = 1

data_clean
```

"carter I will wait for you at fanclub chat.. even tough you're not my favorite ",
'has a mild left inner ear infection.. and its got this irritating blocked feeling since sunday.. ',
'Hey there! Nope. My cuteness is away for awhile ',
"It's only tuesday ",
"Haha not even yo. I just didn't know how to do that to you on twitter Haha",
'Someones alarm clock or a phone woke me up at am...Still got my headache from yesterday night ',
'awww poor puppy is she ok? ',
'i might break down and eat some buffalo wings tomorrow ',

Figure 4.2: Presentación de los datos cargados originales.

Después, al hacer la construcción de nuestro modelo, podemos especificar el procedimiento matemático que utilizaremos. De este modo nuestro modelo seguirá el algoritmo que deseamos siga nuestra red neuronal convolucional. Esto es a partir de `tf.keras.Model`. Una vez creado el modelo, procedemos con el



```

class DCNN(tf.keras.Model):

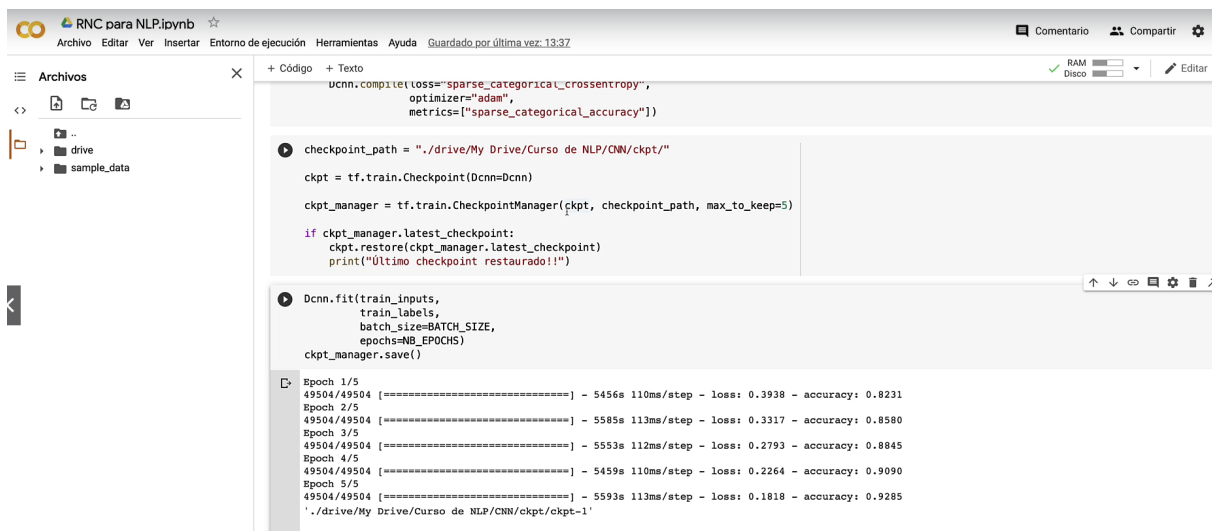
    def __init__(self,
                 vocab_size,
                 emb_dim=128,
                 nb_filters=50,
                 FFM_units=512,
                 nb_classes=2,
                 dropout_rate=0.1,
                 training=False,
                 name="dcnn"):
        super(DCNN, self).__init__(name=name)

        self.embedding = layers.Embedding(vocab_size,
                                           emb_dim)
        self.bigram = layers.Conv1D(filters=nb_filters,
                                     kernel_size=2,
                                     padding="valid",
                                     activation="relu")
        self.trigram = layers.Conv1D(filters=nb_filters,
                                      kernel_size=3,
                                      padding="valid",
                                      activation="relu")
        self.fourgram = layers.Conv1D(filters=nb_filters,
                                       kernel_size=4,
                                       padding="valid",
                                       activation="relu")

```

Figure 4.3: Creación del Modelo matemático (Vectorial).

entrenamiento de nuestra RNC, que en este caso constará de nuestro dataset ya en texto y limpio para optimizar el resultado. Creamos una carpeta donde se guardará nuestro Checkpoint, este consta de 5 capas de entrenamiento.



```

dcnn.compile(loss="sparse_categorical_crossentropy",
             optimizer="adam",
             metrics=["sparse_categorical_accuracy"])

checkpoint_path = "./drive/My Drive/Curso de NLP/CNN/ckpt/"
ckpt = tf.train.Checkpoint(DCNN=dcnn)
ckpt_manager = tf.train.CheckpointManager(ckpt, checkpoint_path, max_to_keep=5)

if ckpt_manager.latest_checkpoint:
    ckpt.restore(ckpt_manager.latest_checkpoint)
    print("Último checkpoint restaurado!!")

dcnn.fit(train_inputs,
        train_labels,
        batch_size=BATCH_SIZE,
        epochs=NB_EPOCHS)
ckpt_manager.save()

```

```

Epoch 1/5
49504/49504 [=====] - 5456s 110ms/step - loss: 0.3938 - accuracy: 0.8231
Epoch 2/5
49504/49504 [=====] - 5585s 113ms/step - loss: 0.3317 - accuracy: 0.8580
Epoch 3/5
49504/49504 [=====] - 5553s 112ms/step - loss: 0.2793 - accuracy: 0.8845
Epoch 4/5
49504/49504 [=====] - 5459s 110ms/step - loss: 0.2264 - accuracy: 0.9090
Epoch 5/5
49504/49504 [=====] - 5593s 113ms/step - loss: 0.1818 - accuracy: 0.9285
'./drive/My Drive/Curso de NLP/CNN/ckpt/ckpt-1'

```

Figure 4.4: Creación del Modelo matemático (Vectorial).

Finalmente la evaluación de nuestra aplicación se basará en tres aspectos. El primero será en tomar un conjunto de datos de nuestros resultados, un conjunto aleatorio de testing para ver la eficiencia de las predicciones emocionales donde la precisión es 0.5711 entorno a un 80 por ciento. Se evalúa con una frase propia como "Eres tan divertido", para ver el porcentaje de felicidad en dicha frase, obteniendo un 88 por ciento y finalmente se realiza la tokenización de las palabras en la frase, en la cual nos da las coordenadas en nuestro conjunto aleatorio.

▼ Evaluación

```
[34] results = Dcnn.evaluate(test_inputs, test_labels, batch_size=BATCH_SIZE)
      print(results)
```

```
500/500 [=====] - 3s 6ms/step - loss: 0.5711 - accuracy: 0.8088
[0.5711000561714172, 0.8088124990463257]
```

```
Dcnn(np.array([tokenizer.encode("You are so funny")])), training=False).numpy()
```

```
array([[0.88222784]], dtype=float32)
```

```
[36] tokenizer.encode("You are so funny")
```

```
[135, 39, 21, 1034]
```

```
[ ]
```

Figure 4.5: Evaluación de nuestra aplicación de Sentimientos con RNC.

4.2 Conclusión

En este proyecto se puede aprender como diferentes áreas de la ciencias de la computación participan en conjunto para resolver un tomador de decisiones en cuestión de procesamiento de lenguaje natural. De este modo por medio de Machine Learning para la limpieza de datos, la inteligencia artificial para el tratado de los datos con librerías como Tensorflow. Las matemáticas en la conversión vectorial por medio de nuestro modelo para el texto y el NLP para la precisión de nuestra aplicación con emociones. En conjunto, podemos crear software bastante robusto e interesante para poder de este modo aportar a la ciencia y poder vender ideas y productor útiles en diferentes áreas.

Las aplicaciones pueden ir desde aplicaciones en comprensión de textos, hasta asistentes para toma de decisiones para personas autistas que no comprendan el impacto de una frase y así saber interpretar una charla o un libro, etc.

Bibliography

- [1] Gemma García. Análisis de sentimiento en los procesos de búsqueda de información en internet, UNIVERSIDAD COMPLUTENSE DE MADRID (Tesis Doctoral), 2020.
- [2] Montañés,R.Aznar,R. Del Hoyo, R. Aplicación de un modelo híbrido de aprendizaje profundo para el análisis de sentimientos en Twitter, CEUR Workshop Proceedings 2172, 2018.
- [3] Aiala Rosa, Luis Chiruzzo, Mathias Etcheverry, Santiago Castro. Análisis de Sentimiento de Tweets en español utilizando SVM y CNN. RETUYT en TASS, 2017.
- [4] Bermejo Escobar, Danitza Yvette; Vizcarra Aguilar, Gerson Waldyr. Modelo basado en aprendizaje profundo para el análisis de sentimiento en español. Universidad nacional del altiplano (Tesis), 2019.
- [5] Alva-Segura, Daniel Abraham. Análisis del Sentimiento Político en Twitter durante las Elecciones Congresales 2020 en el Perú. Universidad Internacional de la Rioja. (Tesis de Maestría), 2021.
- [6] Civera Sancho, Javier. Análisis de sentimiento con aprendizaje profundo en un entorno de Big Data Universidad de Zaragoza, EINA, 2019.
- [7] Maritza Flores Domínguez, José Luis Tapia Fabela. Análisis de sentimiento a nivel de documento en críticas de cine en español. Research in Computing Science 149(8), 2020.