Modelo de procesamiento de lenguaje natural para detectar la tasa de éxito de un artículo sobre otro

PROYECTO DE GRADO

Jonatan Ordóñez Burbano
Yesid Leonardo López Sierra

Asesor
Andrés Alberto Aristizábal Pinzón
PhD

FACULTAD DE INGENIERÍA
MAESTRÍA EN CIENCIAS DE DATOS
SANTIAGO DE CALI
2021

Modelo de procesamiento de lenguaje natural para detectar la tasa de éxito de un artículo sobre otro

Jonatan Ordóñez Burbano
Yesid Leonardo López Sierra

Trabajo de grado para optar al título de Máster en Ciencias de Datos

Asesor
Andrés Alberto Aristizábal Pinzón
PhD

FACULTAD DE INGENIERÍA
MAESTRÍA EN CIENCIAS DE DATOS
SANTIAGO DE CALI
2021

TABLA DE CONTENIDO

Resumen

Muchas personas comparten actualmente noticias, enlaces o videos a familiares y amigos, sin ser conscientes del impacto que pueden tener en las decisiones o forma de actuar de las personas. Un ejemplo claro, que recientemente se ha vivido en Colombia, corresponde al paro nacional que está sucediendo al momento de la entrega de esta tesis. Los colombianos han vivido como las noticias inducen a las masas a tomar decisiones que afectan el ambiente político social y económico del país. También se ha visto como las noticias pueden llegar a generar miedo en el pueblo, o incluso, a desinformarlo en el caso de las noticias falsas. Por estas razones resulta muy importante determinar el impacto que puede tener una noticia.

El problema planteado radica en la inexistencia de una manera de predecir el impacto que puede tener una noticia para una comunidad de usuarios. Por lo tanto, el objetivo consiste en implementar un modelo de aprendizaje automático que permita predecir, con la mejor fidelidad posible, la viralidad de artículos en línea. Para esto se utilizó una metodología enfocada a proyectos de aprendizaje automático denominada CRISP-DM.

Dado que este proyecto fue una propuesta de investigadores en Barcelona, la forma en que se valida este trabajo es mediante una encuesta donde se comprueban los objetivos, hallazgos y resultados alcanzados, versus lo que ellos esperaban.

Finalmente, se obtuvo como mejor resultado aquel correspondiente al modelo donde el núcleo de la arquitectura se basaba en un modelo pre entrenado, denominado BERT, el cual permitía predecir, para una pareja de títulos de noticias, si el primer título sería más viral que el segundo.

Agradecimientos

Damos gracias a Dios por brindarnos la paciencia, constancia y persistencia en aquellos momentos donde más los necesitábamos, también por iluminarnos en el camino correcto.

A nuestros padres por apoyar y fomentar nuestra insaciable hambre de conocimiento.

Agradecimientos a Didier Grimaldi y Carlos Carrasco Ferre, investigadores del ESADE *Business School* y la Universidad La Salle Ramon Llull en Barcelona, por permitirnos hacer parte de un proyecto tan ambicioso y retador, por sus invaluables retroalimentaciones y por motivarnos a mejorar los modelos con sus interesantes ideas y aportes.

A Sebastián García Acosta, por su increíble brillantez, sugerencias, contribuciones y por sus estimulantes ideas para el presente proyecto. Un muchacho que, a pesar de su corta edad, tiene un talento increíble.

Finalmente, pero no menos importante, a nuestro director Andrés Alberto Aristizábal, no solo por su tiempo y aportes en el transcurso de proyecto, sino por mostrarnos que, en los momentos oscuros, la gente buena brilla. Alguien que nos enseña a ser, más que buenos profesionales, excelentes personas.

1. Introducción

1.1 Contexto y antecedentes

En internet no se presenta el contenido de las noticias igual; por ejemplo, aquel contenido que evoca gran cantidad de emociones es más viral (Berger, 2009), y las redes sociales son una fuente de contagio emocional de escala masiva. La sociología y la psicología social definen diez dimensiones para caracterizar las relaciones humanas: "*conocimiento, poder, estatus, confianza, apoyo, romance, similitud, identidad, diversión y conflicto*" (Berger & Milkman, 2012). Además, se tienen resultados preliminares que muestran que estos diez conceptos se expresan a través de lenguaje natural, y, por lo tanto, dan forma a dinámicas observables de interacciones sociales. Sin embargo, se presenta una falta de evidencia del efecto de estas diez dimensiones en las noticias online que son virales (Carrasco, 2020).

Para la presente investigación, se usó como fuente de datos una colección de experimentos realizados por la compañía Upworthy (https://www.upworthy.com/), la cual se dedicada a la publicación de noticias y cuyo objetivo es el análisis y estudio de su influencia, a través de la agrupación de estas en paquetes. Cada paquete contiene un conjunto de noticias junto con sus títulos, imágenes y parte de su contenido. Esta manera de agrupar las noticias hace posible la identificación de las diez dimensiones psicológicas y su relación o efecto sobre la viralidad. Es importante tener en cuenta que, para nuestro objetivo de investigación, la viralidad está definida por la cantidad de clics que recibe una noticia (Matias, 2015).

Al medir el impacto viral en una noticia, se disminuye el tiempo necesario para la publicación de revistas. Sin esta medición, normalmente, el proceso consiste en dos etapas: en la primera se distribuyen las revistas a una cantidad específica de personas para ser sometidas a prueba y en la segunda etapa se analizan los resultados obtenidos en la primera con el objetivo de publicar aquellas revistas que parecen que tendrán éxito.

El proceso mencionado anteriormente toma tiempo e implica una inversión de esfuerzo, y este incide no solo en aquellas personas que seleccionan a las primeras revistas en el primer filtro, sino también el grupo de personas que se encargan de elegir las revistas que consideran llamativas. De esta manera, aquellas agencias o revistas científicas se enfrentan a un proceso tedioso debido a la incertidumbre en términos de viralidad.

1.2 Planteamiento del problema

Actualmente, no existe una forma de predecir el impacto viral que puede llegar a tener una noticia o una revista en una comunidad de usuarios.

1.3 Objetivo general

Implementar un modelo de aprendizaje automático que ofrezca la mejor calidad para detectar la tasa de éxito de un artículo sobre otro

1.4 Objetivos específicos

1. Efectuar una limpieza y depuración del conjunto de artículos base que serán utilizados del conjunto de datos de Upworthy.
2. Proponer dos modelos diferentes de predicción de tasa de éxito de un artículo sobre otro.
3. Realizar un análisis de sentimientos de diez dimensiones a los artículos seleccionados del conjunto de datos de Upworthy.
4. Evaluar los modelos propuestos considerando el mejor resultado de exactitud como la métrica clave de evaluación para detectar, para una pareja de artículos, cuál tendrá más éxito.
5. Seleccionar el mejor modelo que ofrezca los mejores resultados para la detección de éxito sobre parejas de artículos.

1.5 Organización del documento

Este documento se estructura de la siguiente manera: el capítulo 2 describe el marco teórico donde presentan los conceptos en los que se apoya la solución del proyecto. El capítulo 3 expone la metodología que se usó para llevar a cabo el proyecto y para cumplir los objetivos mencionados anteriormente. El capítulo 4 describe la propuesta de la solución que fue llevada a cabo con el proceso detallado; explicando desde el conjunto de datos usado para implementar la solución, hasta las arquitecturas de redes neuronales y sus resultados en las métricas de evaluación. El capítulo 5 describe la encuesta que fue realizada para validar el cumplimiento del objetivo general. El capítulo 6 expone los resultados obtenidos de la solución planteada. Por último, en el capítulo 7 se mencionan las conclusiones del trabajo de grado y el trabajo pendiente que puede ser realizado para continuar con el proyecto.

2. Antecedentes

2.1 Marco teórico

La fundamentación teórica de este trabajo de grado se centra en los conceptos relacionados con el aprendizaje automático, el concepto de aprendizaje secuencia a secuencia, procesamiento de lenguaje natural, redes neuronales recurrentes, LSTM y *Transformers*.

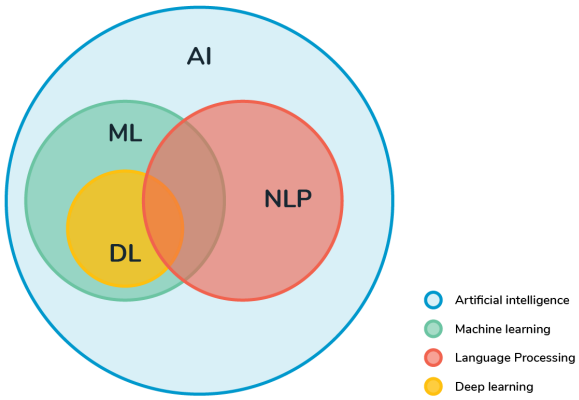


Ilustración 1: NLP, *machine learning* y *deep learning*.

2.1.1 Aprendizaje automático

En el mundo actual se producen grandes volúmenes de datos, y gracias a algoritmos auto entrenados pertenecientes al campo del aprendizaje automático, es posible crear conocimiento y realizar predicciones. Aprendizaje automático es un campo de la inteligencia artificial. Arthur Samuel lo define como: “la ciencia que permite a los computadores aprender, sin ser programados explícitamente”.

2.1.2 Aprendizaje profundo

Este es un campo específico dentro del aprendizaje automático: una nueva forma de asumir las representaciones de aprendizaje de un conjunto de datos que hace énfasis en aprender capas sucesivas de representaciones significativamente incrementales. La palabra “profundo” en su nombre hace referencia a una sucesión de capas de representaciones. (Chollet, 2017)

El aprendizaje profundo hace parte de un conjunto de métodos más amplio, basados en asimilar las representaciones de datos. En general, existen varias definiciones sobre este concepto, pero todas tienen algo en común: múltiples capas de procesamiento no lineal y el

aprendizaje supervisado o no supervisado de representaciones de características en cada capa. (Aprendizaje profundo, s.f.)

2.1.3 Aprendizaje secuencia a secuencia

Es un tipo de aprendizaje que es capaz de transformar una secuencia de elementos en otra secuencia, por ejemplo, la secuencia de palabras en una oración (Vaswani et al., 2017). Estos tipos de modelos son particularmente buenos en problemas relacionados a traducciones. Estos se componen de un decodificador y un codificador (Sutskever et al., 2014). El codificador mapea una secuencia de entrada de representaciones de símbolos $(x_1, ..., x_n)$ a una secuencia de representaciones $z = (z_1, ..., z_n)$. El decodificador se encarga de, dada la secuencia Z, generar una secuencia de salida $(y_1, ..., y_m)$. Este tipo de aprendizaje tiene una ventaja sobre el aprendizaje profundo debido a que, a diferencia de este, no se requiere que la dimensión de las entradas y salidas sean fijas.

2.1.4 Procesamiento de lenguaje natural - NLP

Este tipo de algoritmos les permiten a los computadores procesar y analizar los datos en lenguaje natural para realizar tareas como por ejemplo traducción de imágenes, análisis de sentimientos, generación de nuevo lenguaje natural, etc. Para solucionar exitosamente este tipo de problemas complejos, se debe representar el lenguaje natural de forma que el computador pueda entenderlo (Vasilev, 2019).

Es importante tener en cuenta que el texto, a diferencia de una imagen, se maneja una dimensión en vez de dos, equivalente a una secuencia de palabras larga y única. Además, la estructura del texto tiene varios niveles jerárquicos: caracteres, palabras, oraciones y finalmente, párrafos. Cuando hablamos de imágenes, sabemos que generalmente, los pixeles están relacionados a un objeto, mientras que para las palabras no sabemos si todas se refieren al mismo sujeto (Vasilev, 2019).

2.1.5 Redes neuronales recurrentes - RNN

Las redes neuronales recurrentes, son redes que permiten el procesamiento de datos que requieren un manejo de contexto, a través del almacenamiento de un estado temporal. Por ejemplo, el procesamiento de textos implica que los elementos de una secuencia se relaciones entre una palabra y otra. Por este motivo, es posible usar las RNNs para solucionar tareas asociadas a datos donde el orden secuencial tenga importancia. Ejemplos de la aplicación de este tipo de redes neuronales es la traducción de lenguajes, reconocimiento de voz, predicción del siguiente elemento de una serie de tiempo, etc.

Las RNNs son nombradas de esta manera debido a que cada neuronal procesa secuencialmente un conjunto de datos y a su vez, tienen como entrada el dato de salida de su

predecesor. Por lo tanto, este tipo de modelos tienen memoria sobre el tiempo gracias a que las neuronas preservan información a través de un estado interno. Por esta razón, se dice que las RNNs manejan un estado de recurrencia, ya que básicamente amplían su memoria para aprender de experiencias o información importante que ha pasado hace mucho tiempo.

En las RNN, cada estado es dependiente de todos los estados anteriores a través de su relación de recurrencia. Sin embargo, una de las principales limitantes es que solo son capaces de mirar hacia atrás con unos pocos pasos, llegando al problema del desvanecimiento del gradiente (Vasilev, 2019).

2.1.6 *Long-short term memory* - LSTM

Este tipo de arquitectura permite manejar dependencias a largo plazo debido a una celda de memoria especial. De hecho, este tipo de algoritmo trabaja tan bien que la mayoría de los problemas resuelto de las RNNs han sido gracias a las LSTMs (Hochreiter & Schmidhuber, 1997).

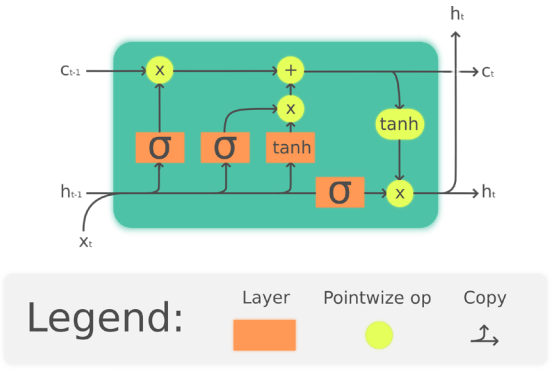


Ilustración 2: Estructura celda LSTM

La principal idea del LSTM es el estado de la celda C donde la información puede ser o explícitamente escrita o removida para que el estado se mantenga constante si no hay interferencia del exterior. Una típica LSTM se compone de 3 compuertas: una compuerta de olvido, una compuerta de entrada y una compuerta de salida.

Gracias a esta celda de memoria, es posible contrarrestar el problema del desvanecimiento del gradiente que se presentaba en la RNNs.

2.1.7 Transformers

Es una arquitectura novedosa que tiene como propósito solucionar tareas de secuencia a secuencia mientras se manejan aquellas dependencias que tienen un mayor alcance con

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

estructura contextual y sintáctica de la oración (Adaloglou, 2020).

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

Ecuación 1: Función para la capa de atención

De la ecuación anterior, Q es una matriz que contiene el *Query* (representación vectorial de una palabra en la secuencia), K son todas las *Keys* (representaciones de vectoriales de todas las palabras en la secuencia) y V son los valores. La anterior es la función que define la capa de atención la cual le dice al modelo a qué palabras debe poner más atención, de esta manera, se tendrá como resultado los pesos finales de atención como una distribución de probabilidad, la cual contiene las palabras contextualizadas y donde el modelo deberá darle más importancia.

2.1.8 Representaciones de codificador bidireccional de *Transformers* – BERT

Esta sigla significa, representación de codificador bidireccional de *Transformers*. Esta es una técnica basada en redes neuronales para el preentrenamiento de NLP. A diferencia de otros modelos donde se hace el procesamiento de manera lineal y que se pone atención en diferentes palabras sin tener en cuenta el contexto, estos modelos hacen el procesamiento de forma bidireccional, es decir, analizan las palabras que están antes y después, y sí tienen en cuenta el contexto (Devlin & Chang, 2018). Adicionalmente, este modelo ayuda a mitigar uno de los más grandes retos de NLP que corresponde a la escasez de datos de entrenamiento, pues este modelo es pre entrenado que usa como *corpus* a Wikipedia.

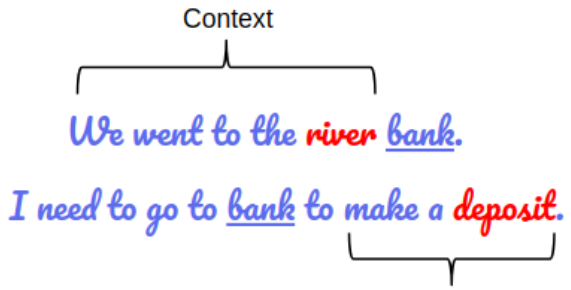


Ilustración 3. Fuente: https://www.inboundcycle.com/blog-de-inbound-marketing/google-bert-que-es-como-funciona

Por lo tanto, este modelo es importante debido a que nos permite generar una representación de cada palabra en el vocabulario teniendo en cuenta el contexto de las otras palabras en la oración

2.2 Estado del arte

Los titulares son una fuente importante para capturar la atención de los lectores e incluso logran influir en su experiencia de lectura. De hecho, aproximadamente seis de cada diez personas limitan su lectura únicamente a los encabezados sin hacer clic en los artículos completos (Piotrkowicz, Dimitrova, Otterbacher, et al., 2017). A pesar de este conocimiento, los titulares no han sido considerados como la única fuente de información para la popularidad de los artículos.

El *procesamiento del lenguaje natural* (NLP por sus siglas en inglés) ha sido un tipo de enfoque que se le ha dado al estudio de los titulares. Esta rama de la inteligencia artificial permite a los computadores entender, interpretar y manipular el lenguaje humano. Este tipo de aprendizaje se ha aplicado a la extracción de información, el reconocimiento de entidades con nombre, la categorización de textos y el etiquetado de partes de la palabra (Olsson, 2009).

En esta sección se presentan tres distintas formas de predecir la popularidad de las noticias.

2.2.1 Predicción y comprensión de la popularidad social de las noticias con características de saliencia emocional

Este artículo principalmente se centra en las propiedades de las noticias socialmente populares con un interés en las emociones trasmitidas a través de los títulos, es decir, principalmente se enfoca en cómo las emociones influyen en las noticias y en extraer las emociones que intensifican a las características. Los datos usados para realizar esta tarea son obtenidos de la API de Facebook de seis editoriales que recibieron más de 17 millones de "me gusta" y "compartir" (Gupta & Yang, 2019).

De esta publicación se concluyó que contrario a lo que se puede pensar que solamente aquellas noticias que evocan un sentimiento negativo son más virales, las publicaciones que tienen puntajes superiores en las diferentes dimensiones también están asociadas a la popularidad (Gupta & Yang, 2019).

Finalmente, se obtiene que el mejor modelo para predecir la popularidad social es un modelo híbrido que se compone de las diferentes características de las emociones, y de un modelo BERT. En este caso, y a diferencia de la propuesta de solución de esta investigación, solamente se enfocan en 5 dimensiones: valentía, alegría, ira miedo y tristeza.

2.2.2 Predicción de popularidad usando modelos tradicionales

Esta investigación tuvo como objetivo encontrar el mejor modelo, y su grupo de características, para predecir la popularidad de noticias online. Además, se buscaba ayudar a las compañías que se dedican a publicar noticias online para que pudieran predecir la popularidad de una noticia antes de su publicación.

Principalmente se centraban únicamente en clasificar usando varios modelos tradicionales como Bosques Aleatorios, Máquinas vectoriales de soporte, Naïve Bayes y redes neuronales entre otros. Los datos usados para entrenar a los modelos provienen de un sitio web llamado *Mashable* (Namous et al., 2019). Sin embargo, para este artículo no se realizó ningún tipo de análisis de sentimientos ni de extracción de características.

Finalmente se concluye que aquellos modelos que obtienen mejores resultados son Arboles Aleatorios y las redes neuronales con una exactitud del 65 % para ambos casos.

2.2.3 Usando los títulos para predecir la popularidad de artículos en línea en Facebook y Twitter

En este artículo se propone y desarrolla la idea de utilizar únicamente los títulos de las noticias para predecir su popularidad, ya que son los títulos los que logran capturar la atención del lector e influir en su experiencia de lectura (Piotrkowicz, Dimitrova, Otterbacher, et al., 2017). Este artículo es particularmente interesante ya que anteriormente no se había considerado el título de una noticia como un elemento crucial en la predicción la popularidad de esta. Para evaluar el modelo de predicción fueron utilizados los títulos encontrados en Facebook y Twitter de las editoriales The Guardian y New York Times.

Como resultado, se concluye que las características extraídas de los títulos tienen un impacto en el desempeño del modelo de predicción, cuando son consideradas por sí solas. Además, que los resultados del modelo de predicción dependen de la fuente de la noticia.

Es importante resaltar de este artículo que se sacaron varias características relativas al título, como por ejemplo las entidades y los sentimientos que evocaban (positivo o negativo). Además, el mejor modelo para consistía en un vector de soporte que se enfocaba en regresión.

A continuación, se observa un cuadro comparativo de los artículos comparados con la solución a implementar:

	Predicción y comprensión de la popularidad social de las noticias con características de saliencia emocional	**Predicción de popularidad usando modelos tradicionales**	**Usando los títulos para predecir la popularidad de artículos en línea en Facebook y Twitter**	**Nuestra solución**
Fuente de los datos	Publicaciones de editoriales de Facebook	Sitio web llamado Mashable	Artículos de noticias de las editoriales en Twitter y Facebook	Upworthy
Cantidad de emociones	Valentía, alegría, ira, miedo y tristeza	No	Positivo o negativo	10 dimensiones sociales
Extracción de entidades	No	No	Si	Si
Usaba solamente el título	Si	Si	Si	Si
Tipo de modelos	NLP: Bert	Modelos tradicionales	Vector de soporte de regresión	NLP: Codificadores de Transformers y BERT
Tipo de tarea	Regresión y clasificación	Clasificación	Regresión	Regresión y clasificación

3. Metodología

3.1 Equipo de trabajo

Este proyecto es realizado por tres personas: Dos estudiantes y el director del proyecto. La siguiente tabla explica la función de cada persona en el proyecto.

Persona	Función	Rol en el proyecto
Yesid Leonardo López	Estudiante. Tiene como función ejecutar las fases de CRISP. Especialmente se enfoca en la fase de modelado y entendimiento de los datos.	Científico de datos: responsable del entendimiento de los datos y de construir los modelos de aprendizaje automático.
Jonatan Ordóñez	Estudiante. Tiene como función ejecutar las fases de CRISP. Especialmente se enfoca en la fase de modelado y exploración de datos.	Científico de datos: responsable del entendimiento de los datos y de construir los modelos de aprendizaje automático.
Andrés Aristizábal Pinzón	Director. Tiene como función orientar y asesorar a los estudiantes en temas relacionados a los modelos predictivos utilizados, la metodología CRISP. Su rol es fundamental para aplicar con éxito la metodología CRISP-DM.	Interesado: orientar a los estudiantes en la aplicación correcta de la metodología y en el análisis de los resultados obtenidos.

Tabla 1: Descripción de las funciones de los integrantes del proyecto.

El tiempo dedicado al proyecto es de 8 horas a la semana por estudiante. Adicionalmente, se realiza una reunión semanal con el director y otros interesados para ver el progreso del proyecto y recibir retroalimentación del trabajo hecho.

3.2 Fases de desarrollo del proyecto

Para realizar este proyecto es fundamental la minería de datos. Por lo tanto, hace conveniente la aplicación del modelo CRISP-DM. El ciclo de vida de esta metodología consta de seis fases, tal como se ve en la **tabla 2.** Las flechas indican las dependencias más importantes y frecuentes entre las fases, el círculo externo simboliza la naturaleza cíclica de la minería de datos.

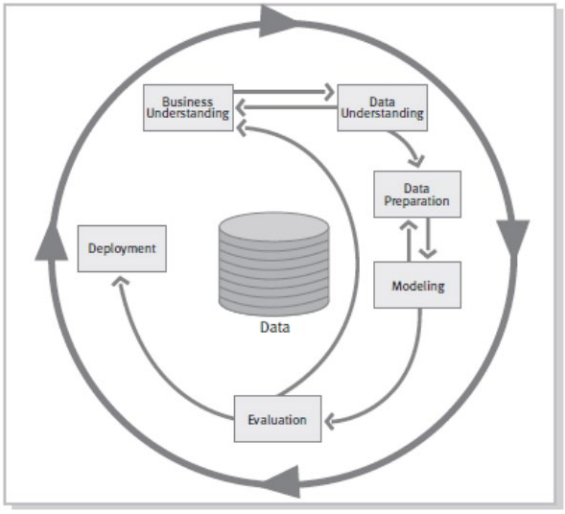


Ilustración 4: Fases del modelo CRISP

En la **tabla** se observan cada una de las fases que tiene el modelo CRISP-DM, con sus correspondientes actividades y entregables:

Fases CRISP-DM	Actividades	Entregables
Entendimiento del negocio	- Fijar los objetivos y requerimientos - Análisis del problema - Definición del problema - Definición de objetivos. - Marco teórico - Estado del arte - Metodología - Cronograma	- Anteproyecto - Diccionario de variables.
Entendimiento de los datos	- Recolección de los datos suministrados por Upworthy. - Describir los datos. - Explorar los datos.	- Exploración de los datos.

Preparación de los datos	- Pre procesar los títulos por palabras, eliminar puntuación, palabras vacías. - Reconocimiento de las entidades en los títulos. - Creación de nuevas variables. Ingeniería de características. - Selección de los datos. - Limpiar los datos para entrenar el modelo de forma correcta.	- Código con la preparación y limpieza de datos
Modelado	- Definición de la arquitectura de la red - Construcción de los modelos para identificar las diez dimensiones. - Construcción de los modelos de regresión. - Entrenamiento del modelo con el conjunto de entrenamiento. - Evaluación de los modelos mediante el conjunto de validación. - Elegir un protocolo de evaluación.	- Marco teórico del anteproyecto. - Archivo binario en Python del modelo de regresión. - Archivo binario que tendrá el protocolo y la evaluación de modelos mediante métricas como R^2 ajustado. - Archivo binario donde se encuentre la implementación de la evaluación.
Evaluación	- Evaluación de los resultados del modelo respecto a las expectativas de los interesados	- Documento con revisión de los resultados obtenidos
Despliegue	- Realizar una presentación a los interesados	- Documento donde se narran los resultados.

Tabla 2: descripción de las actividades a desarrollarse (CRISP-DM)

En la Tabla 3 se mapea cada fase del modelo CRISP-DM con los objetivos específicos:

Fase CRISP-DM	**Objetivo relacionado**
Entendimiento del negocio	Objetivo 1
Entendimiento de los datos	Objetivo 1
Preparación de los datos	Objetivo 1
Modelado	Objetivo 2 Objetivo 3 Objetivo 4
Evaluación	Objetivo 5
Despliegue	Objetivo 5

Tabla 3: fases de CRISP-DM mapeado con los objetivos.

A continuación, se muestra la forma en que está descompuesto cada objetivo en actividades.

Objetivo Específico 1: efectuar una limpieza y depuración del conjunto de artículos base que serán utilizados del conjunto de datos de Upworthy.

Actividades:

1.1 Recolección de los datos suministrados por Upworthy.
1.2 Describir los datos.
1.3 Explorar los datos.
1.4 Pre procesar los títulos por palabras, eliminar puntuación, palabras vacías.
1.5 Reconocimiento de las entidades en los títulos.
1.6 Creación de nuevas variables. Ingeniería de características.
1.7 Selección de los datos.
1.8 Limpiar los datos para entrenar el modelo de forma correcta.

Elicitación de requerimientos

Objetivo Específico 2: proponer dos modelos diferentes de predicción de tasa de éxito de un artículo sobre otro.

Actividades:

2.1 Definición de la arquitectura de la red.

2.2 Construcción de los modelos de regresión.
2.3 Entrenamiento del modelo con el conjunto de entrenamiento.

Objetivo específico 3: realizar un análisis de sentimientos de 10 dimensiones a los artículos seleccionados del conjunto de datos de Upworthy.

Actividades:

3.1 Definición de la arquitectura de la red neuronal.
3.2 Construcción de los modelos de regresión para identificar cada dimensión.
3.3 Entrenamiento del modelo con el conjunto de entrenamiento.

Objetivo específico 4: evaluar los modelos propuestos considerando el mejor resultado de exactitud como la métrica clave de evaluación para detectar, para una pareja de artículos, cuál tendrá más éxito.

Actividades:

4.1 Evaluación de los modelos mediante el conjunto de validación.
4.2 Elegir un protocolo de evaluación.

Objetivo específico 5: seleccionar el mejor modelo que ofrezca los mejores resultados para la detección de éxito sobre parejas de artículos.

5.1 Evaluación de los resultados del modelo respecto a las expectativas de los interesados.
5.2 Realizar una presentación a los interesados.

4. Predicción de viralidad para dos revistas basado en los títulos

4.1 Recolección de los datos

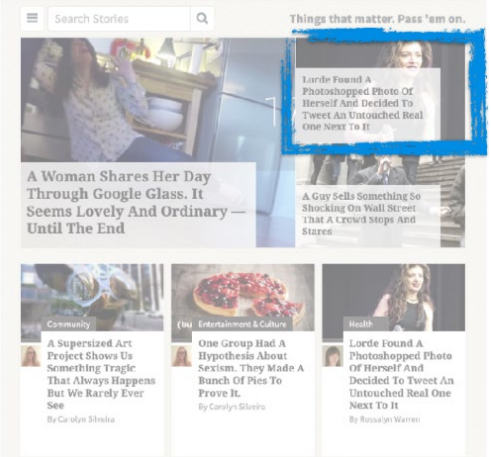


Ilustración 5: A/B testing

Para el desarrollo de este trabajo de grado se utilizó un conjunto de datos suministrado por Upworthy; sitio web dedicado a la narración positiva de noticias que tiene como objetivo generar contenido con impacto positivo tanco social como cultural (Goodinc, n.d.). Estos datos son el resultado de una prueba de *A/B* realizados por la empresa del año 2013 al 2015. Estas pruebas consistían en mandarle a un conjunto aleatorio de usuarios, noticias donde solamente se les mostraba una el título e imagen (como se observa en la **ilustración 6**) y se medía la cantidad de clics que los usuarios daban al ingresar en la noticia.

4.2 Procesamiento de lenguaje natural

En esta sección se menciona las formas en que se hizo el preprocesamiento de las oraciones en los títulos antes de crear las nuevas características. Primero, se aplicó una tokenización a los títulos para convertirlos en unidades más pequeñas tales como palabras. Segundo, se realizó el derivación regresiva *y* lematización de las palabras anteriores con el objetivo de convertir las palabras a su forma base, por ejemplo, cuando se realiza la derivación regresiva a palabras como *viral*, *virality*, *viralization* quedan reducidas a *viral.* Por otro lado, cuando se realiza lematización se tiene en cuenta el análisis morfológico de las palabras, usualmente, requiere *buscar* en un diccionario, por ejemplo, a una palabra como *better,* su lema termina siendo *good* (Manning et al., 2008). Finalmente, se eliminan las palabras de parada, es decir, aquellas palabras sin significado, por ejemplo: artículos, preposiciones, pronombres, etc.

4.2 Ingeniería de características

Este conjunto de datos tenía diferentes columnas que correspondían a metadatos relacionada a una notica, sin embargo, muchos de estos datos correspondía a información que no era mostrada al usuario que daba clic en la noticia (por ejemplo, la descripción, la sentencia de

apertura, entre otros), por lo tanto, del conjunto de datos original, solamente se dejó el título debido a que era lo único que podía tener en cuenta el usuario para abrir la noticia. Desafortunadamente, la información relacionada a las imágenes de las noticias no estaba disponible el conjunto de datos.

Debido a que la solución corresponde a un tipo de aprendizaje supervisado, se usa como variable objetivo la columna *clics*, que corresponde a la cantidad de personas que le dieron clic a la noticia, y la columna de *impressions* que corresponde a la cantidad de personas que recibieron la noticia.

4.2.1 Dimensiones sociales

Una de las tareas realizadas fue crear características que correspondían a las 10 dimensiones mencionadas en el paper de "¿Qué hace al contenido virtual viral?", para esto se utilizaron modelos LSTM y el algoritmo de aprendizaje no supervisado para obtener representaciones vectoriales de palabras (GloVe) de incrustaciones de palabras pre entrenadas. Como resultado se obtienen diez modelos que se encargan de darle un valor a cada título, en cada dimensión. De esta manera, se tenía que para un título como "*5 reasons you may need to plan a vacation*" se tenían valores de 0 a 1 para cada una de las diez dimensiones, donde la dimensión con el valor más alto era diversión y un puntaje cercano a cero para romance.

4.2.2 Análisis de sentimientos

Para cada título del conjunto de datos, se identificó qué tan positivo o negativo era el sentimiento asociado. Para realizar esta tarea de análisis de sentimientos, se utilizó un modelo pre entrenado llamado ***flair*** el cual daba como resultado un valor de 0 a 1 para saber qué tan positiva es la noticia.

4.2.3 Verbos, adjetivos, entidades y cantidades asociadas

Por otro lado, mediante el título, y la librería de código abierto de *NLP* llamada SpaCy, utilizada comúnmente para realizar extracción de la información o para preprocesamiento de texto, se produjeron diferentes características por cada título: entidades, verbos, adjetivos, cantidades de verbos y cantidad de adjetivo.

4.2.4 Tasa de clics

Ya que uno de los enfoques propuestos es implementar modelos de clasificación para identificar cuál de los títulos es más viral, se puede obtener el porcentaje de viralidad definido como la cantidad de clics sobre el total de personas que recibieron la noticia. Posteriormente, se puede decir que un artículo fue viral cuando supere un umbral definido.

4.2 Arquitecturas de la solución

Como se menciona en los objetivos específicos, esta investigación se encamina a responder la pregunta mediante los dos tipos de aprendizaje supervisado: regresión y clasificación.

Clasificación: esta tarea consiste en la predecir mediante un modelo, si el primero, de dos títulos dados, tendrá más clics. **Regresión:** esta tarea consiste en estimar mediante un modelo la diferencia relativa entre el número de clics de dos títulos. Esta diferencia relativa es calculada como: $|h_1 - h_2|/h_{max}$, donde h_1 y h_2 corresponde al número de clics del primer y segundo título respectivamente, y h_{max} corresponde al número máximo que un título puede tener.

4.2.1 Primera arquitectura – LSTM bidireccional junto con todas las características para predecir la diferencia relativa de clics

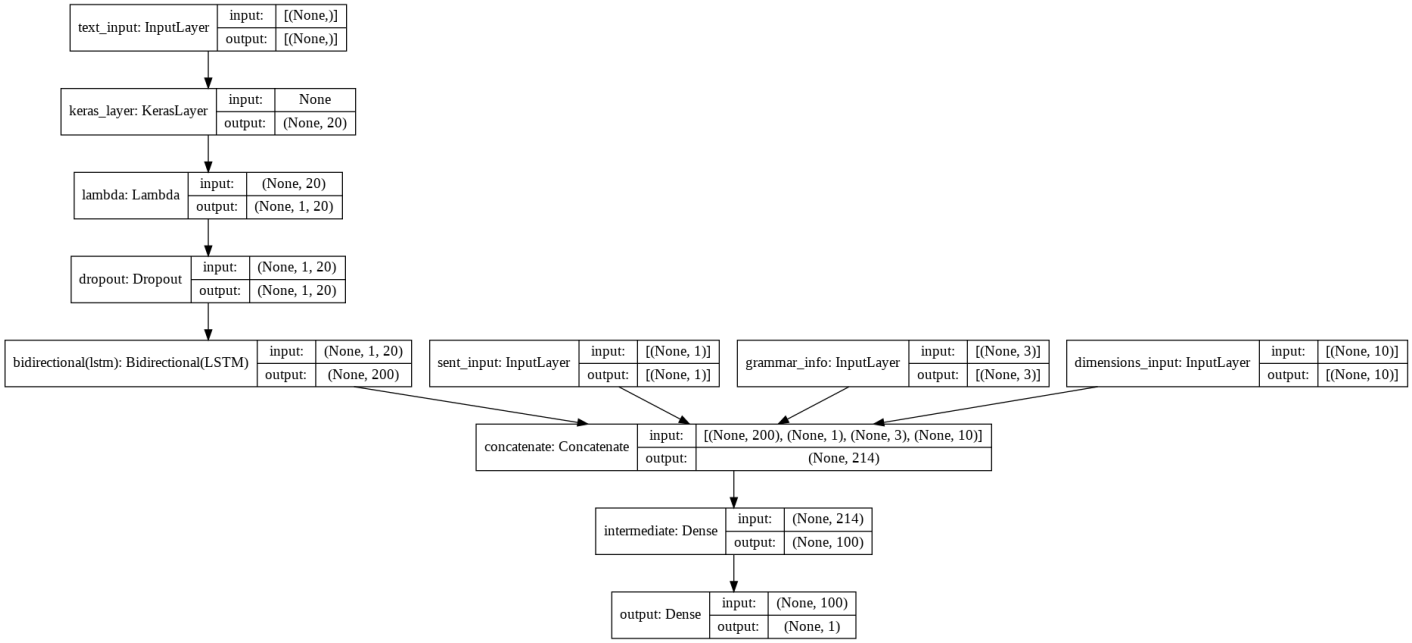


Ilustración 6: arquitectura primera solución

Como se observa en la **ilustración**[1]**,** esta arquitectura tiene 4 ramas donde la primera rama recibe el título una vez pasa por la fase de transformación de texto, donde es unida a una capa de abandono para minimizar el sobreajuste y posteriormente a una LSTM bidireccional, la segunda rama es una capa que maneja los sentimientos (positivos o negativos), la tercera

[1] Para ver en más detalle la arquitectura dirigirse a la sección de anexos

rama corresponde a la información gramática (tales como pronombres, adjetivos, verbos y sus respectivas cantidades), y la cuarta y última rama corresponde a la capa que maneja las 10 dimensiones. Finalmente, las 4 ramas se unen con una capa que las concatena y pasa por una capa densa que maneja una función relu (y la cual es usada típicamente para clasificar) que estima la diferencia relativa de clics entre los títulos. Esta arquitectura se descartó porque, a pesar de hacer un ajuste a los hiper parámetros, en todos los casos se encontró que daba como resultado un valor negativo (o en algunos casos muy cercano a cero) para el coeficiente de determinación.

4.2.2 Segunda arquitectura – Modelos de atención con BERT para regresión

Este enfoque consiste en un modelo que recibe dos títulos como entrada ordenados de forma descendente por el número de clics he intenta estimar la diferencia relativa usando una capa de atención. Las consultas de esta capa de atención son secuencias de incrustación de palabras producidas por BERT para el primer título, por otro lado, sus llaves y valores son las incrustaciones de palabras producidos por BERT para el título que tiene la menor cantidad de clics.

La idea detrás de esta segunda propuesta consiste en que el mecanismo de atención es capaz de encontrar las llaves correctas para cada consulta, así, cuando se tengan títulos más virales, el modelo tratará de poner más atención a aquellas palabras que también están presentes en el título que es menos viral como una manera de encontrar que lo hace más viral que el otro. A pesar de esto, el modelo no resulto con un buen desempeño y, al igual que el primer enfoque de regresión, no se tuvo una métrica positiva para el coeficiente de determinación.

Debido a que, en lo realizado anteriormente, se presentan valores negativos para la métrica del coeficiente de determinación, los modelos de clasificación quedan descartados para predecir la cantidad diferencia relativa entre el número de clics de dos títulos.

4.2.3 Tercera arquitectura – Codificador *Transformer* para clasificación

Es interesante observar para este enfoque, que, debido a que es un modelo el cual es creado desde cero con un codificador *transformer*, adicionalmente, se observan las relaciones que se crean a partir de las cabezas de atención del *transformer* (***ilustración***)

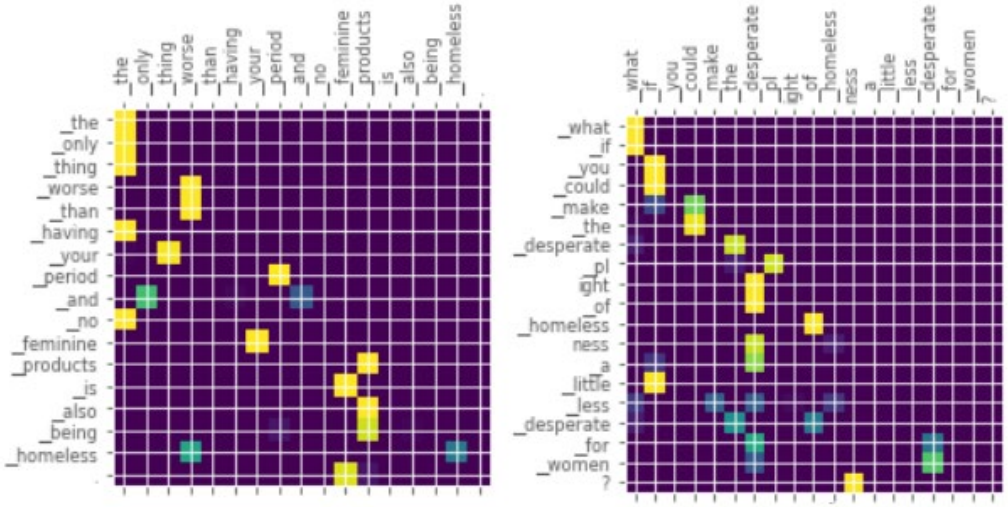
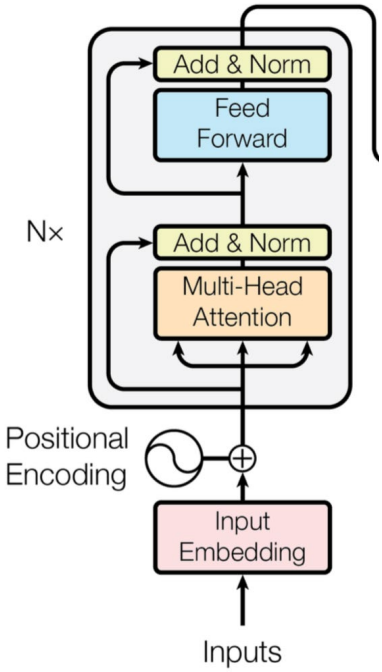


Ilustración 7: relaciones entre palabras para un título.



lustración 8: Codificador tomado de “attention is all you need” por Vaswani et al.

En la **ilustración 9,** se muestra una ilustración del codificador utilizado para esta arquitectura. La idea detrás de este es poder darles contexto a las incrustaciones de palabras.

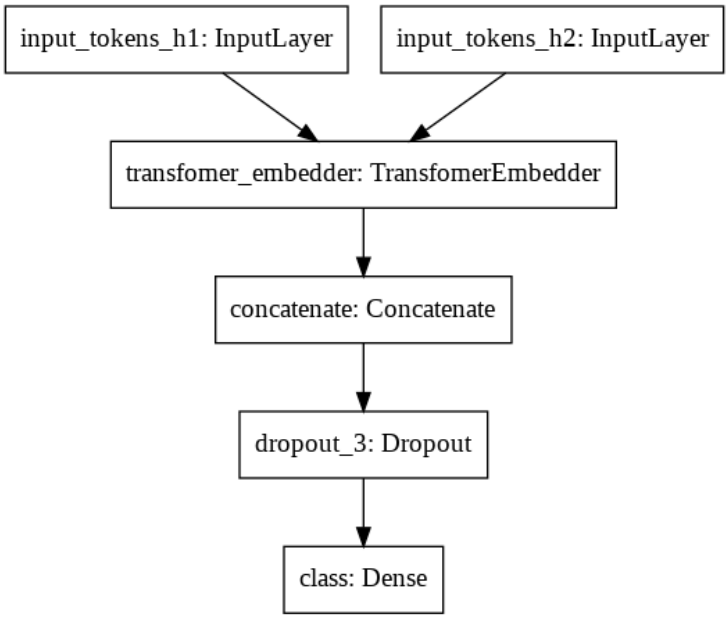


Ilustración 9: Arquitectura usando el codificador *Transformer*.

Como se puede observar en la **ilustración 9**, se reciben como entrada dos tokens que corresponden a las cabezas de atención que tiene el modelo, posteriormente, se pasa al codificador y posteriormente se pasa a una capa de concatenación debido a que el codificador dará como resultado dos inputs para cada cabeza de atención, se tiene una capa de abandono para prevenir el sobreajuste y finalmente se pasa a una capa densa con una función de activación tipo sigmoide para que realice la clasificación.

4.2.4 Cuarta arquitectura – BERT para clasificación

Para esta arquitectura se pasa de crear el codificador desde cero con las incrustaciones de palabras, a utilizar un modelo pre entrenado llamado BERT que fue creado por Google, este permite dar sentido a las palabras mediante un contexto bidireccional.

En la **ilustración**[2] se puede observar la arquitectura de la cuarta solución:

[2] Para revisar la arquitectura con un mejor detalle dirigirse a los anexos.

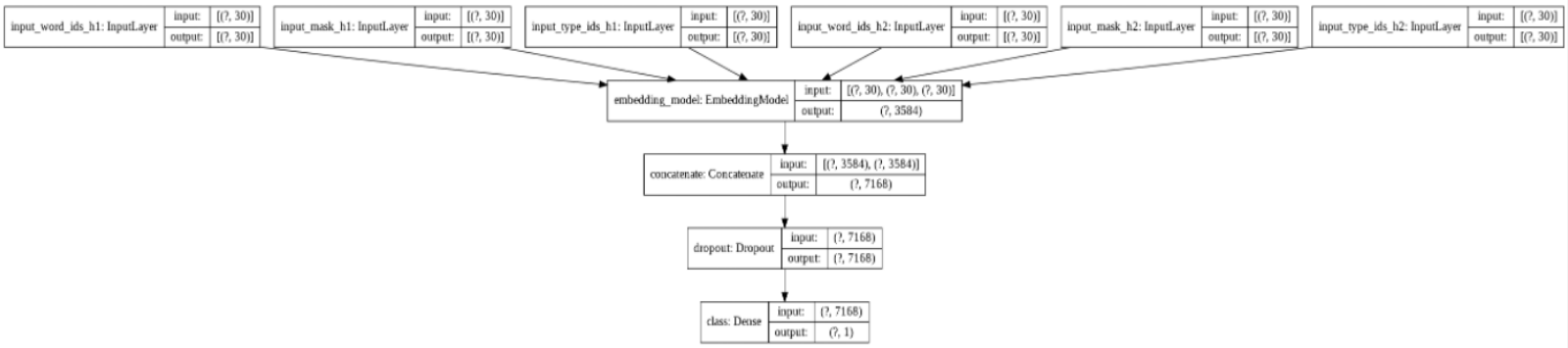


Ilustración 10: Arquitectura del modelo que usa BERT

De la ilustración anterior se tienen 6 ramas iniciales: las 3 primeras corresponden a *words_ids, mask y type_ids* del título 1 y las restantes a *words_ids, mask y type_ids* del título 2. Posteriormente se pasa a una capa que se encarga de generar las incrustaciones de palabras contextualizadas para posteriormente añadir una capa de concatenación con el objetivo de unir todos los outputs de BERT, luego una capa de abandono para disminuir el sobreajuste, y finalmente una capa densa que tiene la función sigmoide como función de activación, esta última capa dirá si el primer título tendrá más clics que el segundo.

5. Diseño de experimento de validación

Para validar los resultados obtenidos de la propuesta, primero, se realizó una presentación a las personas interesadas donde se mostró un resumen de resultados del proyecto, posteriormente, se creó una encuesta, la cual fue enviada a aquellos que auditaban el proyecto. Al ser este trabajo una propuesta por investigadores del ESADE *Business School* y la Universidad La Salle Ramon Llull en Barcelona, y no hay forma de ponerla a prueba con datos reales de producción, se utilizó la encuesta como mecanismo de validación.

Encuesta de Validación de la propuesta de la solución

Esta encuesta es realizada con el objetivo de evaluar la satisfacción de la solución propuesta para el proyecto de *"Modelo de procesamiento de lenguaje natural para predecir la viralidad de artículos en línea considerando las emociones y las dinámicas sociales"*.

Concepto	Sí	No
¿La solución cumplió con las expectativas?	X	
¿Las métricas obtenidas en el mejor modelo (64% de exactitud) son aceptables para usted?	X	
¿Considera que hay trabajo futuro para continuar al proyecto?	X	
¿Considera que los hallazgos hechos en el proyecto son de utilidad?	X	
¿Está de acuerdo en solo usar la solución de clasificación debido a los bajos resultados en la regresión (valores inferiores a cero en el coeficiente de determinación) como se discutió con el equipo?	X	

Qué comentarios le gustaría dejar respecto al proyecto

Los resultados son buenos: 64%. Podríamos mirar cómo mejorarlos buscando patrones de las 100 mejores parejas y las 100 peores parejas.

Ilustración 11: respuesta de los investigadores que auditaban el proyecto

La anterior fue la respuesta dada por los dos auditores del proyecto, en este caso, ambos decidieron llenar una encuesta posterior a una discusión de resultados. Como se puede observar, los resultados obtenidos del proyecto fueron satisfactorios para ambos. Además, se

valida que el enfoque usado para la solución del proyecto fue la clasificación. Finalmente, se confirma que el mejor modelo utilizado es la arquitectura relacionada a BERT.

Por otro lado, como mencionan en los comentarios, se resalta que como trabajo futuro se pueden revisar aquellos resultados negativos y tratar de encontrar patrones en ellos, de tal manera que posteriormente pueda mejorarse el modelo.

6. Resultados obtenidos

En esta sección se mostrarán los resultados obtenidos para las 4 arquitecturas propuestas anteriormente: la arquitectura con todas las características generados, además del título y la arquitectura que utilizaba BERT para regresión. La arquitectura que usaba un codificador *transformer* y la arquitectura que usaba BERT para clasificación.

6.1. Resultados Regresión

Tabla 4: resultados de los modelos después del *afinamiento de hiper parámetros*

Modelo	**Coeficiente de determinación**
LSTM	-4.75
BERT	≈ 0

Como se observa en la tabla anterior, para BERT, se obtiene un mejor coeficiente de determinación y solamente se usó el título como entrada. Sin embargo, para ambos casos, se obtuvieron resultados negativos, o muy cercanos a cero, a pesar de una exhaustiva búsqueda de los mejores hiper parámetros y configuraciones de la arquitectura. Por lo tanto, este enfoque de regresión fue descartado.

6.2. Resultados Clasificación

Para la tercera arquitectura, la cual tenía como objetivo predecir si el primer título sería más viral que el segundo (para parejas de títulos), se encontró que era mejor, en términos de exactitud, usar solamente el título y no todas las características.

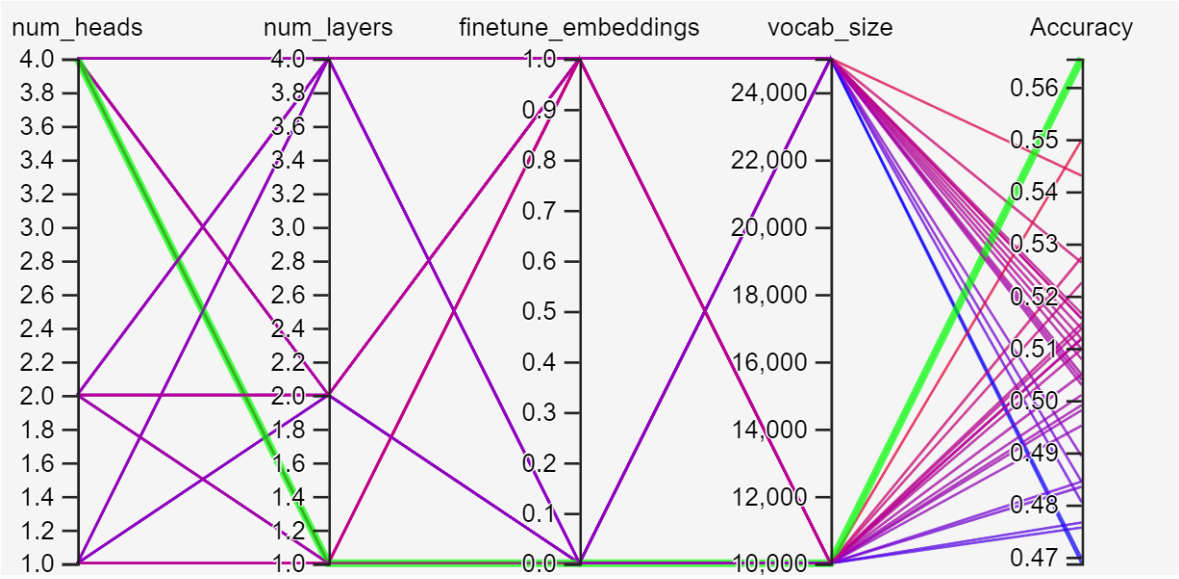


Ilustración 12: afinamiento de hiper parámetros para modelo que usa el codificador *transformer*.

En la anterior imagen[3] se puede observar el resultado de aplicar afinamiento de hiper parámetros al modelo. Aquí se observa que, al realizar varias configuraciones, el mejor resultado corresponde a la línea verde que maneja 4 cabezas de atención, 1 capa, con un valor de 0 para el *fine-tune* de las incrustaciones de palabras y un tamaño de vocabulario de 10.000, consiguiendo una exactitud del 56,5 %.

El resultado anterior da un valor un poco mejor respecto a la línea base (teniendo en cuenta que este es del 50%).

Posteriormente, utilizando la arquitectura que contiene BERT para clasificación, se obtienen mejores resultados:

[3] Para revisar en mejor detalle los resultados puede dirigirse al siguiente enlace:

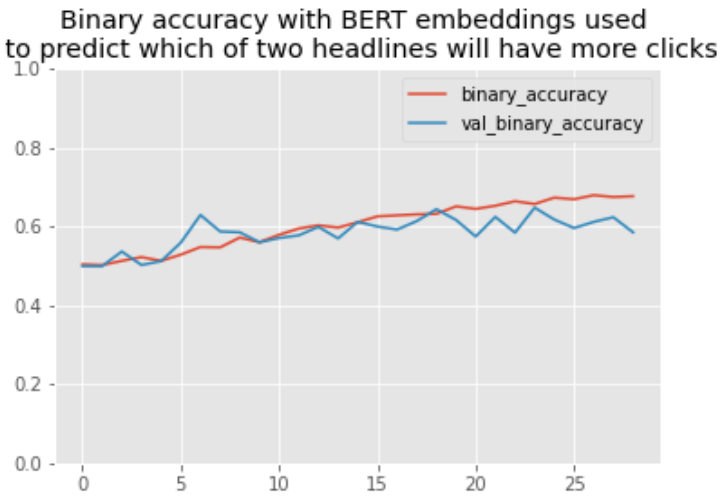


Ilustración 13: Correctitud vs Cantidad de épocas.

Como se observa en la ilustración, el mejor resultado se da con 7 épocas, arrojando una exactitud del 64 %. Este resultado termina siendo un valor mucho más alto que la tercera arquitectura y es significativamente mejor que la línea base (50 %).

A continuación, se presenta una tabla comparativa con los resultados relacionados al problema de clasificación:

Tabla 5: resumen de resultados de clasificación.

Modelo	**Exactitud**
Baseline	50 %
Arquitectura con *transformer encoder*	56.5 %
Arquitectura con BERT	**64 %**

Como se observa en la tabla anterior, la arquitectura con BERT es la que tiene mejor exactitud (*accuracy*), por lo tanto, esta se define como **la mejor arquitectura para el modelo** de la solución.

```
Headlines:
h1: the oscars have been singing this song to themselves for decades. she's singing it out loud.
h2: neil patrick harris is a tv and broadway star hosting the oscars. she's suggesting a musical number.
h1 clicks: 28
h2 clicks: 17
Raw prediction: [[0.83906513]]
Raw target: [1]
Intuitive predicton: will have more clicks
Intuitive target   : will have more clicks
```

Ilustración 14: Ejemplo del comportamiento del modelo

En la ilustración anterior se muestra un ejemplo con dos títulos, se da la predicción (*prediction*) y el valor esperado (*target*). Aquí se puede observar que el primer título va a ser más viral que el segundo, con una exactitud del 84%, lo cual es correcto (según el resultado esperado).

7. Conclusiones y trabajo futuro

La dificultad para predecir el impacto de una noticia puede tener grandes implicaciones a nivel social, económico y político. Un ejemplo claro de esta problemática se evidencia con el reciente paro nacional, que inició a finales de abril del 2021 en Colombia, donde el poder de las redes sociales, las noticias y su impacto viral ha sido un catalizador de los acontecimientos. Por esta razón, se corrobora que medir el impacto de una noticia puede resultar vital para una sociedad. Es aquí donde el objetivo del proyecto cobra sentido al buscar implementar un modelo de aprendizaje automático que permita predecir la viralidad de noticias.

Después de una exhaustiva investigación, donde se utilizaron dos enfoques: regresión, para predecir la diferencia relativa de clics para una pareja de títulos, y clasificación, para identificar si el primer título tendrá más clics que el segundo, se establecieron las siguientes conclusiones.

Para el caso de la regresión, pese a haber implementado dos modelos, el primero utilizando todas las características obtenidas a partir del proceso de la ingeniería de características y un modelo LSTM, y el segundo, haciendo uso de un modelo pre entrenado BERT que tan solo utilizaba el título como entrada, no se obtuvieron los resultados esperados y por lo tanto se descartaron. Para este enfoque se puede concluir que no fue posible encontrar la diferencia relativa entre los títulos, esto pudo ser influenciado debido que se trabajó con pocos datos pues usábamos un conjunto pequeño, suministrado para implementar las arquitecturas. Además, la imposibilidad de calcular la tendencia (debido a la librería de terceros que bloqueaba por cantidad de peticiones) para el momento en que se publica la noticia, es una característica importante que pudo haber dado más información al modelo, y presentar mejores resultados. Es importante resaltar que estimar la cantidad de clics es una tarea mucho más compleja que identificar cuál de las noticias será más viral.

En el caso de la clasificación se alcanzaron resultados satisfactorios, utilizando dos tipos de modelos. Por un lado, aquel que solamente usaba el decodificador del modelo de *transformer* y por otro, uno que hacía uso de BERT para dar contexto a las palabras. Aunque ambos presentaron resultados de exactitud superiores a la línea base, fue este último, que utilizaba BERT, el que alcanzó el mayor valor. Para este, se puede intuir que el preentrenamiento de BERT, con el gran conjunto de datos de wikipedia, y el mecanismo de atención que le da a las palabras el contexto, son fundamentales para mejorar la exactitud.

Con base en el desarrollo del modelo, se puede resaltar que la principal contribución hecha en términos de viralidad corresponde a la posibilidad de identificar, para una pareja de títulos, cuál de ellas será más viral. Es decir, se puede evidenciar que la forma en que se redacta el

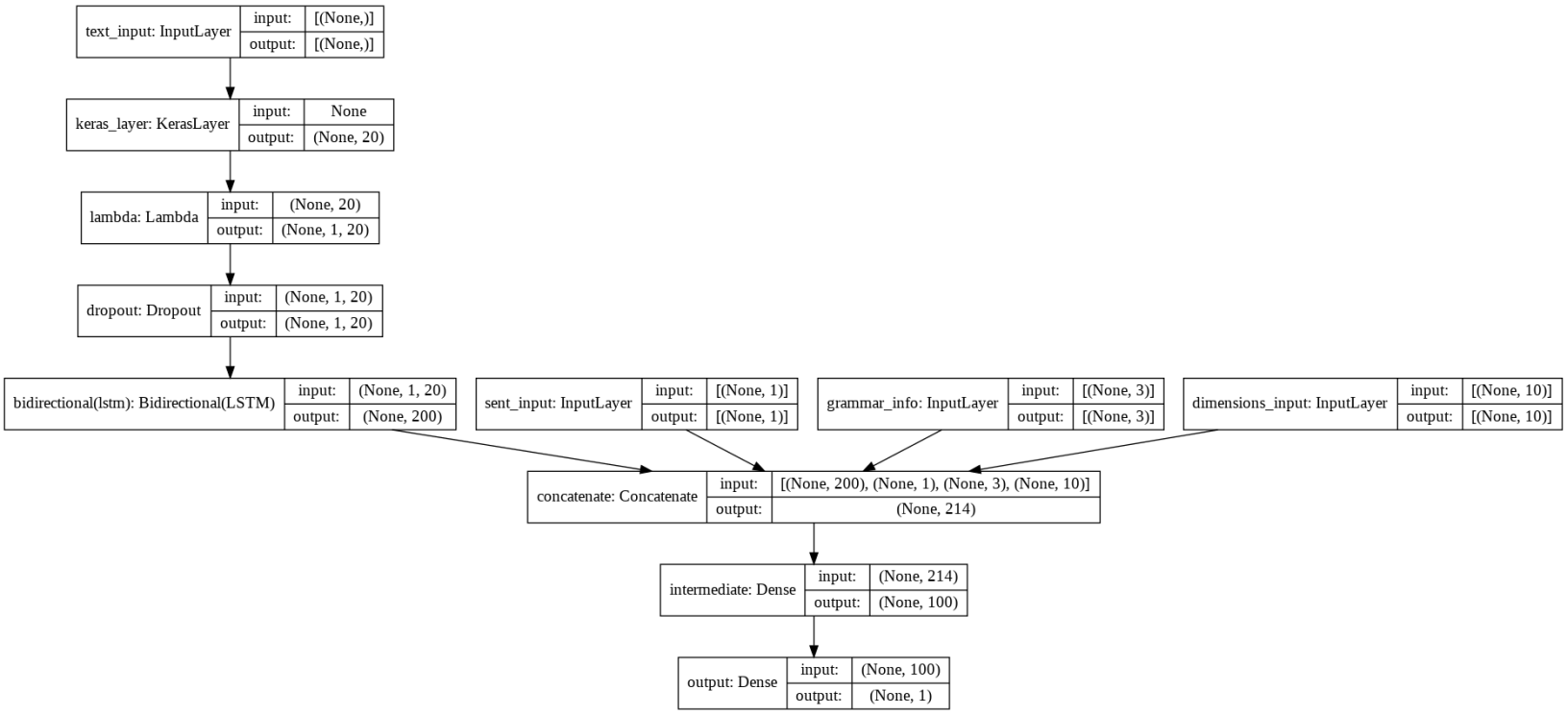
título de la noticia puede impactar, de manera positiva o negativa, la aceptación por parte de la comunidad.

Por otro lado, en términos de trabajo futuro se plantean las siguientes ideas:

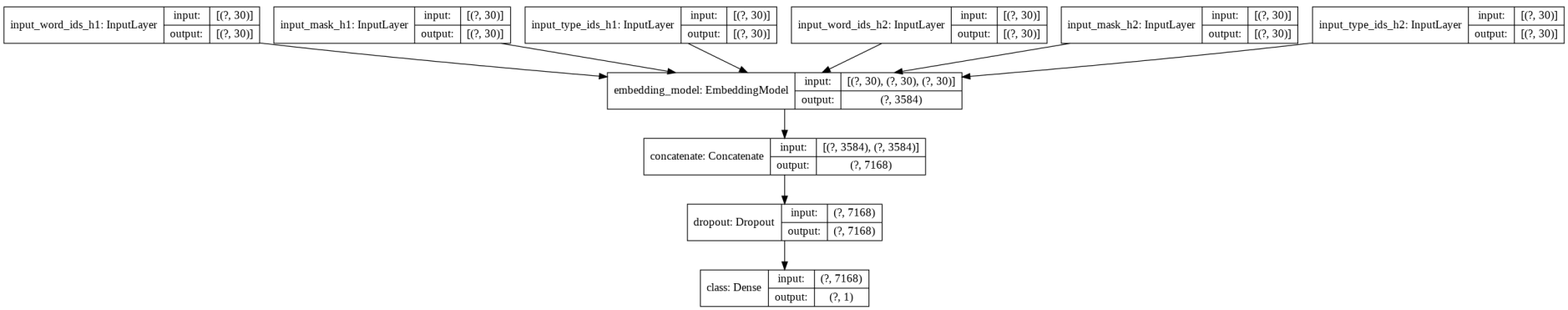
- Identificar las razones o factores que hacen que un título sea más viral, puede ser una idea interesante para entender mejor la viralidad en las noticias. Sin embargo, hay que tener en cuenta que estos tipos de modelos, que usan redes neuronales, tienden a ser cajas negras, a diferencia de otros modelos como árboles de decisión, por lo cual es mucho más complicado interpretarlos y resulta todo un reto encontrar las razones por las cuales el modelo llega a un resultado en particular.
- Además, poder encontrar una forma de capturar las tendencias del momento, para una noticia en específico, y utilizarla como entrada al modelo, mejorará significativamente la correctitud del modelo, e incluso, permitirá encontrar la diferencia relativa para la pareja de clics. Sin embargo, la principal barrera es evitar que la IP sea bloqueada por la cantidad de peticiones que hace por consulta de las tendencias por cada título (en el caso de la librería de terceros llamad *pytrends*).
- Adicionalmente, una propuesta que surge luego de la presentación hecha a los investigadores es la de revisar los títulos donde el modelo se equivoca, para identificar la razón por la que puede fallar este y así poder mejorarlo.
- A mediados de mayo del 2021, Google, la misma empresa que crea BERT, propone un nuevo modelo llamado MUM (siglas para *Multitask Unified Model*), este modelo es la evolución a la arquitectura de BERT y promete ser "1.000 veces más potente" (Nayak, 2021). Por lo anterior, se plantea probar con este nuevo modelo con el objetivo de hallar mejores resultados.
- Finalmente, a partir del proceso y los resultados presentados de este trabajo de grado, se establece una propuesta para la elaboración y futura publicación de un artículo científico en conjunto con los investigadores del ESADE *Business School* y la Universidad La Salle Ramon Llull en Barcelona.

8. Anexos

Primera arquitectura LSTM para regresión: a continuación, se muestra la arquitectura verticalmente para que pueda ser vista de una forma más amigable:



Cuarta arquitectura BERT: a continuación, se muestra la arquitectura verticalmente para que pueda ser vista de una forma más amigable:


input_word_ids_h1: InputLayer
input: [(?, 30)]
output: [(?, 30)]
input_mask_h1: InputLayer
input: [(?, 30)]
output: [(?, 30)]
input_type_ids_h1: InputLayer
input: [(?, 30)]
output: [(?, 30)]
input_word_ids_h2: InputLayer
input: [(?, 30)]
output: [(?, 30)]
input_mask_h2: InputLayer
input: [(?, 30)]
output: [(?, 30)]
input_type_ids_h2: InputLayer
input: [(?, 30)]
output: [(?, 30)]
embedding_model: EmbeddingModel
input: [(?, 30), (?, 30), (?, 30)]
output: (?, 3584)
concatenate: Concatenate
input: [(?, 3584), (?, 3584)]
output: (?, 7168)
dropout: Dropout
input: (?, 7168)
output: (?, 7168)
class: Dense
input: (?, 7168)
output: (?, 1)

Plantilla de la encuesta entregada para validar la solución

Encuesta de Validación de la propuesta de la solución

Esta encuesta es realizada con el objetivo de evaluar la satisfacción de la solución propuesta para el proyecto de "*Modelo de procesamiento de lenguaje natural para predecir la viralidad de artículos en línea considerando las emociones y las dinámicas sociales*".

Concepto	Sí	No
¿La solución cumplió con las expectativas?	X	
¿Las métricas obtenidas en el mejor modelo (64% de exactitud) son aceptables para usted?	X	
¿Considera que hay trabajo futuro para continuar al proyecto?	X	
¿Considera que los hallazgos hechos en el proyecto son de utilidad?	X	
¿Está de acuerdo en solo usar la solución de clasificación debido a los bajos resultados en la regresión (valores inferiores a cero en el coeficiente de determinación) como se discutió con el equipo?	X	

Qué comentarios le gustaría dejar respecto al proyecto

Los resultados son buenos: 64%. Podríamos mirar cómo mejorarlos buscando patrones de las 100 mejores parejas y las 100 peores parejas.

_____

_____

_____

_____

Ilustración 15: encuesta de validación

Referencias Biblográficas

Allsop, D. T. (2007). *Word-of-Mouth Research: Principles and Applications*.

Aprendizaje profundo. (s.f.). Obtenido de wikipedia: https://es.wikipedia.org/wiki/Aprendizaje_profundo

Asch, S. E. (1956). *Studies of Independence and Conformity: A Minority of One Against a Unanimous Majority*.

Berger, J. A. (2009). *What Makes Online Content Viral?*

Chollet, F. (2017). *Deep Learning With Python.*

Rashka, S. (2015). *Python Machine Learning.*

Travers, M. (Marzo de 2020). *Facebook Spreads Fake News Faster Than Any Other Social Website, According To New Research*. Obtenido de Forbes: https://www.forbes.com/sites/traversmark/2020/03/21/facebook-spreads-fake-news-faster-than-any-other-social-website-according-to-new-research/?sh=4d15b8106e1a

Vasilev, I. (2019). *Advanced Deep Learning with Python.*

Adaloglou, N. (2020). *How Transformers work in deep learning and NLP: an intuitive introduction*. https://theaisummer.com/transformer/

Berger, J., & Milkman, K. L. (2012). What makes online content viral? *Journal of Marketing Research*, *49*(2), 192–205. https://doi.org/10.1509/jmr.10.0353

Carrasco, C. (2020). *How emotions and social dynamics drive online virality*. 1–5.

Devlin, J., & Chang, M.-W. (2018). *Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing*. https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

Goodinc. (n.d.). *Upworthy*. About Us. https://goodinc.com/

Gupta, R. K., & Yang, Y. (2019). Predicting and understanding news social popularity with emotional salience features. *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia*, 139–147. https://doi.org/10.1145/3343031.3351048

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. https://doi.org/10.1017/CBO9780511809071

Matias, N. (2015). *Data in the Upworthy Research Archive*. https://upworthy.natematias.com/about-the-archive

Namous, F., Rodan, A., & Javed, Y. (2019). Online News Popularity Prediction. *ITT 2018 - Information Technology Trends: Emerging Technologies for Artificial Intelligence*, *Itt*, 180–184. https://doi.org/10.1109/CTIT.2018.8649529

Nayak, P. (2021). *MUM: A new AI milestone for understanding information*. https://blog.google/products/search/introducing-mum/

Olsson, F. (2009). *A literature survey of active machine learning in the context of natural language processing*. *T2009*(06). http://soda.swedish-ict.se/3600/1/SICS-T--2009-06--SE.pdf

Piotrkowicz, A., Dimitrova, V., & Markert, K. (2017). Automatic extraction of news values from headline text. *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of the Student Research Workshop*, 64–74. https://doi.org/10.18653/v1/e17-4007

Piotrkowicz, A., Dimitrova, V., Otterbacher, J., & Markert, K. (2017). Headlines matter: Using headlines to predict the popularity of news articles on Twitter and Facebook. *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017*, *May*, 656–659.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, *4*(January), 3104–3112.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Transformer: Attention is all you need. *Advances in Neural Information Processing Systems 30*, *Nips*, 5998–6008.