Análisis del Modelo LSTM Mediante Procesamiento De Lenguaje Natural Machine Learning Aplicados A La Detección De Fake News.

Bustinza-Inofuente, Claudio 1

claudio.bustinza@upeu.edu.pe 1

*EP. Ingeniería de Sistemas, Facultad de Ingeniería y Arquitectura, Universidad Peruana Unión*



**RESUMEN**

El presente artículo tiene como objetivo analizar el desempeño del modelo LSTM (Long Short-Term Memory) en la detección de noticias falsas (fake news) en español. Se acabo un estudio previo sobre el uso de LSTM y técnicas de procesamiento de lenguaje natural (NLP) para la detección de fake news, evidenciando resultados diversos en términos de precisión según el conjunto de datos y la configuración utilizados. Se explican aspectos importantes como el preprocesamiento de textos, la creación de word embeddings y el entrenamiento supervisado de redes neuronales. En la experimentación se entrenó un modelo LSTM con datos etiquetados en español, logrando una precisión de 97% en la clasificación de noticias verdaderas y falsas. Se concluye que las técnicas de NLP y deep learning son prometedoras para la detección automática de fake news, pero se requiere mayor investigación para encontrar la combinación óptima de algoritmos y representaciones textuales que maximicen la precisión.

Palabras clave: Detección de Fake News, LSTM, Procesamiento de Lenguaje Natural, Machine Learning.

**ABSTRACT**

The objective of this article is to analyze the performance of the LSTM (Long Short-Term Memory) model in detecting fake news in Spanish. A previous study was carried out on the use of LSTM and natural language processing (NLP) techniques for the detection of fake news, showing varying results in terms of accuracy depending on the data set and settings used. Important aspects such as text preprocessing, creating word embeddings, and supervised training of neural networks are explained. In the experiment, an LSTM model was trained with data labeled in Spanish, achieving an accuracy of 97% in the classification of true and false news. It is concluded that NLP and deep learning techniques are promising for automatic detection of fake news, but more research is required to find the optimal combination of algorithms and textual representations that maximize accuracy.

Keywords: Fake News detection, LSTM, Natural Language Processing, Machine Learning.



**1.- INTRODUCCIÓN**

Las noticias falsas, Fake News en inglés es un problema que aqueja a nuestra sociedad y es uno de los fenómenos más conocidos y extendidos de desinformación. Tal y como deﬁne (Molina, 2021), Las redes sociales se han convertido en un escaparate de importancia relativa. Uno de los éxitos de las redes es la facilidad de comunicación y el impacto en distintas actividades que obedecen a los intereses individuales y grupales (Beatriz et al., 22020), El año 2020 marcó un punto de inflexión, con cifras impactantes que reflejan la magnitud de este problema. En plataformas de redes sociales como Twitter, se registraron asombrosas 47 millones de tuits que compartían información falsa o engañosa, mientras que en Facebook se generaron 1,200 millones de interacciones en sitios web que difundieron Fake News Según lo señalado en (Kemp, 2021). Este incremento en la difusión de noticias falsas no se limitó solo a la cantidad, sino que abarcó una amplia variedad de temas, desde cuestiones políticas hasta, alcanzando niveles sin precedentes de participación y viralidad (Goldstein, 2021). En este contexto, (Escobar D & Quevedo A, 2021) Al realizar una valoración del estado actual de los desarrollos implementados y las investigaciones elaboradas al respecto, se obtuvieron distintos resultados donde se han acertado múltiples soluciones al problema de detección de Fake News aplicando el modelo: Regresión Logística, Arboles de Decisión, Redes Neuronales como: LSTM mediante Procesamiento de Lenguaje Natural y Machine Learning. La idea de evaluar este modelo permite una aproximación clara y pronta a una solución viable.

El problema de la desinformación afecta a las relaciones y a la conﬁanza entre las personas, pues, tal y como aﬁrma Marc Amorós, “una sociedad con mala salud informativa vive condenada a la ceguera y si llegamos al punto en que no podemos conﬁar en las noticias, solo nos creeremos las que reaﬁrmen nuestro pensamiento” (García, 2018).

Escobar Cerron, D. F., & Quevedo Cortes, A. D. (2021). En su trabajo titulado "Modelo prototipo de inteligencia artificial basado en procesamiento de lenguaje natural y redes de neuronas artificiales para la detección de noticias falsas en español", el objetivo fue desarrollar y comparar dos enfoques para determinar la veracidad de noticias en español. El primer enfoque empleó Redes de Neuronas LSTM y NLP, revelando una precisión de entrenamiento limitada, mientras que el segundo enfoque se basó en el clasificador probabilístico Naive Bayes Multinomial, logrando una alta precisión del 98% en pruebas y 99% en entrenamiento. Estos resultados evidenciaron desafíos en el uso de LSTM en textos cortos y traducidos. Al concluir, se logró crear un prototipo de IA con más del 98% de precisión en la detección de noticias falsas o verdaderas.

Escobar Cerron, D. F., & Quevedo Cortes, A. D. (2022). En su investigación titulada "Inteligencia Artificial como alternativa en la detección de noticias falsas," concluyeron que, debido a la presencia de las noticias falsas en diversos ámbitos informativos y su impacto en la opinión pública, es crucial continuar desarrollando algoritmos de Aprendizaje Automático para su detección. A pesar de los avances, este problema persiste y afecta la democracia y la reputación de individuos y organizaciones. Para abordar esta tendencia, plantearon la importancia de implementar mecanismos confiables y precisos, donde la Inteligencia Artificial desempeña un papel crucial al ofrecer modelos y algoritmos efectivos para afrontar este desafío en curso.

Valenzuela, A., & Alonso, J. C. (2020). En su investigación "Detección de Fake News mediante técnicas de Deep Learning," obtuvieron resultados que consisten en 9 modelos de clasificación de textos distintos, los cuales también son aplicables en otros ámbitos de la Inteligencia Artificial. Realizaron un análisis comparativo de estos modelos, evaluando sus ventajas y desventajas, y la mejora de los hiperparámetros de la red neuronal y así obteniendo un valor F1-Score del 98% de efectividad para la detección contra las noticias falsas, destacando así la eficacia de estas técnicas de Deep Learning.

**2.- MATERIALES Y METODOS**

En este estudio, se llevó a cabo una investigación sobre el modelo LSTM para la detección de noticias falsas (fake news) utilizando diversas bibliotecas y técnicas de procesamiento de lenguaje natural (PLN).

## **2.1.- Instrumentos**

Python fue el lenguaje principal utilizado para el desarrollo del modelo. Bibliotecas de Procesamiento de Lenguaje Natural (PLN): Se emplearon NLTK preprocesamiento del texto, incluyendo tokenización, lematización y eliminación de stop words. Bibliotecas de Machine Learning y Deep Learning: TensorFlow y Keras se utilizaron para construir y entrenar el modelo LSTM. El desarrollo se llevó a cabo en Google Colab, que proporcionan un entorno interactivo y acceso a recursos de computación en la nube.

|  |
| --- |
| from tensorflow.keras.preprocessing.text import Tokenizer  from tensorflow.keras.preprocessing.sequence import pad\_sequences  from tensorflow.keras.models import Sequential  from tensorflow.keras.layers import Dense, Embedding, LSTM, Conv1D, MaxPool1D  from sklearn.model\_selection import train\_test\_split  from sklearn.metrics import classification\_report, accuracy\_score  print("Todos los módulos se han importado correctamente.") |

**2.2.- Participantes**

El equipo compuesto por solo un miembro se encargó de evaluar el rendimiento del modelo utilizando métricas de precisión, recall, F1-score y precisión. Se interpreto los resultados y compararon el rendimiento del modelo LSTM con otros enfoques de machine learning. Se diseñó y entrenó el modelo LSTM. seleccionando los hiperparámetros adecuados, implementando la arquitectura del modelo y realizando múltiples experimentos para ajustar y optimizar el rendimiento del modelo. Utilizando bibliotecas como TensorFlow y Keras para construir y entrenar el modelo, y herramientas como Google Colab para aprovechar recursos de GPU en la nube.

**2.3.- Modelo LSTM**

Estructura del Modelo: El modelo LSTM consistió en una capa de embeddings para convertir palabras en vectores de características, seguida de dos capas LSTM con 128 unidades cada una, y una capa densa final con activación sigmoide para la clasificación binaria. La tasa de aprendizaje se fijó en 0.001, utilizando el optimizador Adam.

|  |
| --- |
| model = Sequential()  model.add(Embedding(vocab\_size, output\_dim=DIM, weights = [embedding\_vectors], input\_length=maxlen, trainable=False))  model.add(LSTM(units=128))  model.add(Dense(1, activation='sigmoid'))  model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['acc']) |

Los datos se dividieron en 70% para entrenamiento, 15% para validación y 15% para prueba. El tamaño del lote se fijó en 64 y el modelo se entrenó durante 20 épocas. Se utilizó una técnica de dropout con una tasa de 0.5 para prevenir el sobreajuste. Las métricas utilizadas para evaluar el modelo incluyeron precisión, precisión, recall y F1-score de 97%.

|  |
| --- |
| print(classification\_report(y\_test, y\_pred))  precision recall f1-score support  0 0.98 0.97 0.98 5851  1 0.97 0.98 0.97 5374  accuracy 0.98 11225  macro avg 0.98 0.98 0.98 11225  weighted avg 0.98 0.98 0.98 11225 |

#### **2.4.- Datos Utilizados**

#### Para la investigación se utilizo un conjunto de datos de noticias verdaderas y falsas obtenidas de un repositorio de GitHub, los datos fueron recopilados y etiquetados previamente, permitiendo una clasificación precisa entre noticias. Los artículos varían en longitud, con un promedio de 500 palabras por artículo. Las fuentes incluyen una variedad de medios de comunicación con diferentes orientaciones políticas y temáticas. Etiquetado: Las noticias fueron etiquetadas manualmente por verificadores de hechos y, en algunos casos, utilizando algoritmos de verificación automática basados en el contenido y la credibilidad de las fuentes.

**2.5.- Fuente de los Datos**

Los datos fueron extraídos del repositorio de GitHub disponible en <https://github.com/laxmimerit/fake-real-news-dataset>. Este repositorio contiene una colección extensa de artículos de noticias, debidamente etiquetados como verdaderos o falsos.

**2.6.- Organización del Conjunto de Datos**

La organización de datos incluye varias columnas, las mas relevantes para la investigación fueron el título de la noticia, texto, fecha y categoría. El conjunto de datos está compuesto por un total de 44,908 artículos, de los cuales 21,418 son noticias verdaderas y 23,490 son noticias falsas.

**2.7.- Preprocesamiento de los Datos**

**2.7.1.- Limpieza de Texto:** Se realizaron varias etapas de preprocesamiento para limpiar y preparar los datos antes de utilizarlos en el modelo. Esto incluye la eliminación de caracteres especiales.

|  |
| --- |
| ps.remove\_special\_chars('this ,. @ # is gre@t')  ('this is gret') |

**2.7.2.- Tokenización y Vectorización:** Los textos fueron tokenizados y convertidos en secuencias numéricas mediante técnicas de vectorización, lo que permite su procesamiento por algoritmos de Machine Learning.

|  |
| --- |
| tokenizer = Tokenizer()  tokenizer.fit\_on\_texts(X)  embedding\_vectors = get\_weight\_matrix(w2v\_model) |

**3.- Resultados y Discusión**

En la experimentación que se realizó, se entrenó un modelo LSTM con un dataset de 44,908 noticias tanto falsas como con datos de noticias reales. Se aplicaron técnicas de procesamiento de texto como tokenización, lematización y derivación. El modelo LSTM obtuvo una precisión de 97% en la clasificación de noticias verdaderas y falsas.

**3.1.- Resultado 1**

**3.1.1.- Rendimiento del Modelo**

El modelo LSTM mostró un rendimiento excepcional en la tarea de clasificación, para la detección se utilizó la Función booleana,

Esta función opera en palabras formadas por los valores binarios 0 y 1, que corresponden a "falso" y "verdadero" respectivamente, y devuelve como resultado también los valores binarios 0 o 1.

|  |
| --- |
| real['class'] = 1  fake['class'] = 0 |

Este código extendido realiza las siguientes acciones:

1. Toma el texto original "Noticia falsa" y lo somete a procesamiento.
2. Muestra la predicción del modelo y la clasificación resultante.
3. Incorpora más ejemplos de noticias, algunas potencialmente falsas y otras potencialmente verdaderas, para evaluar el modelo.
4. Procesa cada uno de estos ejemplos adicionales y muestra los resultados obtenidos.

|  |
| --- |
| x = ['Noticia falsa']  x = tokenizer.texts\_to\_sequences(x)  x = pad\_sequences(x, maxlen=maxlen) |

Para evaluar la efectividad del modelo LSTM en la detección de noticias falsas, se llevaron a cabo pruebas utilizando datos de muestra y se analizaron los resultados de las predicciones. El proceso de predicción y la interpretación de los resultados se pueden ilustrar con el siguiente ejemplo:

En este caso, el modelo recibe como entrada una secuencia de texto que ha sido preprocesada (x) y produce una predicción. Esta predicción es inicialmente una probabilidad que varía entre 0 y 1. Luego, se convierte esta probabilidad en una decisión binaria, donde se asigna un valor booleano (falso o verdadero). Posteriormente, este resultado booleano se convierte en un entero, siendo 0 para falso y 1 para verdadero.

El resultado específico “array([[0]])” indica que el modelo clasificó la entrada como una noticia falsa (0). La velocidad de predicción registrada (150ms/step) subraya la eficiencia del modelo en el procesamiento de nuevas entradas, un aspecto crucial para aplicaciones que requieren respuestas en tiempo real.

Además, se examinó la estructura de los datos de prueba (X\_test):

|  |
| --- |
| (model.predict(x) >=0.5).astype(int)  1/1 [==============================] - 0s 150ms/step  array([[0]]) |
| X\_test  array([[ 0, 0, 0, ..., 4, 406, 2575],  [ 0, 0, 0, ..., 493, 175, 20147],  [ 0, 0, 0, ..., 216, 175, 1097],  ...,  [ 0, 0, 0, ..., 197, 175, 4125],  [ 0, 0, 0, ..., 147, 786, 1458],  [ 0, 0, 0, ..., 112, 1288, 1379]], dtype=int32) |

Esta matriz contiene secuencias de texto que han sido codificadas numéricamente y tienen padding aplicado. Cada fila representa una noticia del conjunto de prueba, y cada columna corresponde a una palabra o token en la secuencia. Los números en la matriz son índices que reflejan las palabras en el vocabulario del modelo. Cuando el texto no alcanza la longitud deseada, se completa con ceros. El padding asegura que todas las secuencias tengan la misma longitud para el procesamiento uniforme por parte del modelo.

Los valores no nulos indican palabras específicas dentro del texto de cada noticia. La variabilidad en los últimos números de cada fila (por ejemplo, 2575, 20147, 1097) sugiere una diversidad en el vocabulario y el contenido de las noticias analizadas. Esta representación numérica facilita que el modelo LSTM procese eficientemente el texto de las noticias, capturando patrones lingüísticos y semánticos cruciales para distinguir entre noticias verdaderas y falsas. La capacidad del modelo para manejar estas secuencias complejas y variadas contribuye significativamente a su alto rendimiento en la tarea de clasificación de noticias.

**3.1.2.- Análisis de Predicciones del Modelo**

Para comprender mejor el funcionamiento del modelo LSTM en la detección de noticias falsas, se realizaron pruebas con dos ejemplos contrastantes. Este proceso ilustra cómo el modelo procesa y clasifica diferentes tipos de noticias.

Se define una noticia de entrada como una cadena de texto. La función tokenizer.texts\_to\_sequences(x) convierte el texto en una secuencia de números, donde cada número representa una palabra específica en el vocabulario del modelo asegura que la secuencia tenga una longitud fija (maxlen), añadiendo ceros al principio si es necesario genera una probabilidad de que la noticia sea verdadera, convierte esta probabilidad en una clasificación binaria: 0 para falsa, 1 para verdadera.

**Ejemplo 1:** Noticia Potencialmente Falsa

**Resultado:** El modelo clasifica esta noticia como falsa (0). El tiempo de procesamiento fue de 323ms.

**Análisis:** La noticia sobre una ciudad de oro en la Amazonía contiene elementos típicos de noticias sensacionalistas o falsas. El modelo ha captado correctamente estos patrones y la ha clasificado como falsa.

|  |
| --- |
| x = ['Un grupo de exploradores ha anunciado el descubrimiento de una ciudad perdida hecha completamente de oro en lo profundo de la selva amazónica de Brasil.']  x = tokenizer.texts\_to\_sequences(x)  x = pad\_sequences(x, maxlen=maxlen)  (model.predict(x) >=0.5).astype(int)  1/1 [==============================] - 0s 323ms/step  array([[0]]) |

**Ejemplo 2:** Noticia Potencialmente Verdadera

**Resultado:** El modelo clasifica esta noticia como verdadera (1). El tiempo de procesamiento fue de 137ms.

**Análisis:** Esta noticia contiene información específica sobre procesos judiciales y políticos, con nombres y detalles concretos. El modelo la ha clasificado como verdadera, probablemente debido a su estilo formal y la presencia de información verificable.

|  |
| --- |
| x = ['El Poder Judicial ha decidido devolver a la Fiscalía la acusación contra el expresidente Alejandro Toledo por los delitos de lavado de activos y colusión, en el marco de las investigaciones por el caso Interoceánica Sur, Tramo 4.']  x = tokenizer.texts\_to\_sequences(x)  x = pad\_sequences(x, maxlen=maxlen)  (model.predict(x) >=0.5).astype(int)  1/1 [==============================] - 0s 137ms/step  array([[1]]) |

**Enlace del repositorio GitHub:** [**https://github.com/ClaudioB12/Fake-news-con-LSTM**](https://github.com/ClaudioB12/Fake-news-con-LSTM)

**3.1.3.- Ventajas del Modelo LSTM**

El modelo LSTM demostró ser particularmente efectivo en el manejo de secuencias de texto largas, una característica crucial para el análisis de noticias completas. La capacidad del modelo para procesar tanto los títulos como el texto completo de las noticias permitió una clasificación más precisa, aprovechando toda la información disponible en cada artículo.

**3.1.4.- Limitaciones y Consideraciones**

Mientras que estos resultados son prometedores, es importante reconocer que dos ejemplos no son suficientes para una evaluación exhaustiva. El modelo podría enfrentar desafíos con noticias más ambiguas o con técnicas de desinformación más sofisticadas.

1. Aunque el conjunto de datos es extenso, podría no capturar completamente la evolución constante de las noticias falsas y sus técnicas de desinformación.
2. La efectividad del modelo en temas muy recientes o emergentes podría requerir evaluación adicional, dado que el conjunto de datos tiene una fecha de corte específica.
3. El impacto de la fecha de publicación en la clasificación de noticias podría explorarse más a fondo para entender cómo cambian los patrones de desinformación con el tiempo.

**3.2.- Modelos de Red Neuronal Recurrente (RNN)**

Los modelos de Red Neuronal Recurrente (RNN) muestran una menor susceptibilidad a la longitud de las noticias, ya que transforman cada documento en un vector de tamaño fijo. Aunque la longitud máxima establecida para cada noticia en la RNN recorta solo un pequeño porcentaje del corpus textual, la posible pérdida de información es mínima. La RNN compensa esta desventaja gracias a dos elementos clave: su memoria a corto plazo (LSTM), que preserva información por más tiempo, y la capa Bidireccional, que permite analizar secuencias en ambas direcciones (de principio a fin y viceversa) según Alonso (2020).

En cuanto al uso de Machine Learning se han tomado como inspiración los conocidos “The 7 Steps of Machine Learning” (“Los 7 Pasos del Machine Learning” en español), de Yufeng Guo, que se resumen en los siguientes puntos (Alonso, 2020):

1. Adquisición de los datasets o conjunto de datos

2. Preparación de los datos

3. Elección del modelo

4. Entrenamiento del modelo

5. Evaluación del modelo

6. Ajuste de parámetros

7. Predicción

**3.3.- Preprocesamiento de Datos en el Contexto de LSTM y PLN**

En esta sección se describe el proceso de preprocesamiento aplicado a las oraciones de los títulos antes de la creación de nuevas características. Inicialmente, se llevó a cabo la tokenización de los títulos para convertirlos en unidades más pequeñas, como palabras. Posteriormente, se aplicaron la derivación regresiva y la lematización a las palabras con el propósito de llevarlas a su forma base. La derivación regresiva redujo palabras relacionadas, como viral, virality, y viralization, a su forma más básica, en este caso, "viral". En contraste, la lematización consideró el análisis morfológico de las palabras, buscando en un diccionario, y por ejemplo, convirtiendo la palabra "better" a su lema "good" (De Grado et al., 2021).

**3.4.- Discusión de Resultados**

**Capacidad de Discernimiento:** El modelo demuestra una notable capacidad para distinguir entre noticias potencialmente falsas y verdaderas basándose en el contenido, estilo y contexto del texto. Esto sugiere que el entrenamiento ha sido efectivo en capturar las sutilezas lingüísticas y semánticas que diferencian las noticias falsas de las verdaderas.

**Eficiencia Computacional:** Los tiempos de procesamiento (323ms y 137ms) indican que el modelo es suficientemente rápido para aplicaciones en tiempo real, como la verificación de noticias en plataformas de redes sociales o sitios web de noticias.

**Aprendizaje de Patrones:** La correcta clasificación de estos ejemplos contrastantes sugiere que el modelo ha aprendido patrones lingüísticos, contextuales y estructurales relevantes para la detección de noticias falsas. Esto incluye la identificación de lenguaje sensacionalista, la valoración de la especificidad de la información y el reconocimiento de estilos periodísticos legítimos.

**4.- CONCLUSIÓN**

Los resultados obtenidos en este estudio demuestran el potencial del modelo LSTM junto con técnicas de PLN para la detección automática de noticias falsas en idioma español. Al entrenar el modelo con un conjunto de datos de 44,908 noticias etiquetadas, se logró una precisión del 97% en la clasificación correcta de noticias verdaderas y falsas, lo cual representa un desempeño muy prometedor, es importante reconocer que la efectividad del modelo LSTM puede verse influenciada por diversos factores, como la calidad y tamaño del conjunto de datos de entrenamiento, las técnicas específicas de preprocesamiento textual aplicadas, el método de representación vectorial de palabras (word embeddings), así como la arquitectura y configuración de hiperparámetros de la red neuronal. Si bien los resultados obtenidos son alentadores, es necesario continuar investigando y optimizando estos aspectos para maximizar aún más la precisión del modelo, particularmente en idiomas diferentes al inglés, donde los recursos y conjuntos de datos suelen ser más limitados. Para futuras investigaciones, sería valioso explorar enfoques híbridos que combinen el modelo LSTM con otras técnicas de aprendizaje automático y procesamiento de lenguaje natural, con el objetivo de aprovechar las fortalezas complementarias de diferentes métodos. Además, la creación de conjuntos de datos más grandes, diversos y de alta calidad en español podría contribuir significativamente a mejorar el rendimiento del modelo en la detección de noticias falsas en nuestro idioma.

**REFERENCIAS**

Beatriz, M., Loranca, B., Vanessa, M., Estrada, M., González Velázquez, R., Salomón, D., & Mendoza, L. (s. f.). Procesamiento de Lenguaje Natural: una solución para detectar noticias falsas sobre la 4T en México Natural Language Processing: A solution to Detect Fake News about 4T in Mexico. En *Research in Computing Science* (Vol. 149, Número 8).

Obando Molina, F. A. J. 2021. (2021). *IMPLEMENTACIÓN DE CLASIFICADOR DE NOTICIAS EN IDIOMA ESPAÑOL PARA LA IDENTIFICACIÓN DE FAKE NEWS MEDIANTE EL ANÁLISIS, TRADUCCIÓN AUTOMÁTICA Y VALIDACIÓN DE UN CONJUNTO DE DATOS EN INGLÉS, Y EL USO DE TÉCNICAS DE APRENDIZAJE MÁQUINA Y PROCESAMIENTO DE LENGUAJE NATURAL.* <https://repositorio.ucsm.edu.pe/handle/20.500.12920/10902>

Pérez, C. R. (2019). No diga fake news, di desinformación: una revisión sobre el fenómeno de las noticias falsas y sus implicaciones. *Comunicación*, *40*, 65–74. <https://doi.org/10.18566/COMUNICA.N40.A05>

Simeone, O. (2018). A Very Brief Introduction to Machine Learning with Applications to Communication Systems. *IEEE Transactions on Cognitive Communications and Networking*, *4*(4), 648–664. <https://doi.org/10.1109/TCCN.2018.2881442>

Goldstein, A. (2021). Social media engagement with deceptive sites reached record highs in 2020. Technical report, The German Marshal Fund of the United States. <https://www.gmfus.org/news/social-media-engagement-deceptive-sites-reached-record-highs-2020>

Shu, K., Wang, S., Lee, D., & Liu, H. (2020). Mining disinformation and fake news: Concepts, methods, and recent advancements. In \_Lecture Notes in Computer Science\_ (pp. 1-19). Springer International Publishing. https://doi.org/10.1007/978-3-030-43796-3\_1

International Federation of Journalists. (2022). ¿Qué son las fake news?; Guía para combatir la desinformación en la era de posverdad. <https://www.ifj.org/fileadmin/user_upload/Fake_News_-_FIP_AmLat.pdf>

Rouhiainen, L. (2018). \_Inteligencia artificial\_. Alienta Editorial. <https://planetadelibrosec0.cdnstatics.com/libros_contenido_extra/40/39308_Inteligencia_artificial.pdf>

Bonden, M. A. (2016). \_Inteligencia artificial\_. Turner Publicaciones. <https://books.google.es/books?hl=es&lr=&id=LCnYDwAAQBAJ&oi=fnd&pg=PT3&dq=(Margaret+A:+Bonden,+Inteligencia+Artificial,+Turner+Publicaciones+S.L)&ots=drYuvTbIqc&sig=BRyffpvd0BX17M_ADDr6jPict7A#v=onepage&q=(Margaret%20A%3A%20Bonden%2C%20Inteligencia%20Artificial%2C%20Turner%20Publicaciones%20S.L)&f=false>

IBM. (2020). \_Machine learning\_. https://www.ibm.com/mx-es/analytics/machine-learning

Ortega Riveros, J. A., & Quintero Perozo, D. Y. (2020). Detección automática de noticias falsas en español con técnicas de machine learning (Trabajo de grado). Universidad de los Andes. <https://repositorio.uniandes.edu.co/server/api/core/bitstreams/2b28b6e2-ec23-4e0e-9449-1941b8bedde3/content>

Espejel-Rivera, M. A., Calderón-Suárez, R., Ortega-Mendoza, R. M., Camacho-Bello, C. J., & Máquez Vera, M. A. (2022). Detección automática de noticias falsas usando representaciones textuales tradicionales y soluciones basadas en aprendizaje profundo. \_Pädi: Boletín Científico de Ciencias Básicas e Ingenierías del ICBI\_, \_10\_(3), 120-127. <https://doi.org/10.33062/ap.v3i4.9008>

Zeroual, I., & Lakhouaja, A. (2018). Data science in light of natural language processing: An overview. \_Procedia Computer Science, 127\_, 82-91. <https://doi.org/10.1016/j.procs.2018.01.101>

Molina Abellán, J. (2018). \_Diseño y desarrollo de una interfaz para la generación de lenguaje natural\_ (Trabajo Fin de Grado). Universidad de Alicante, España. <http://hdl.handle.net/10045/80390>

Medrano, J. (2020). Agentes inteligentes para recuperación de información y analítica visual en big data. \_XXII Workshop de Investigadores en Ciencias de la Computación\_. <http://sedici.unlp.edu.ar/handle/10915/103444>

Celi-Parraga, R. J., Varela-Tapia, E. A., Acosta-Guzmán, I. L., & Montaño-Pulzara, N. R. (s.f.). \_Técnicas de procesamiento de lenguaje natural en la inteligencia artificial conversacional textual.\_ \_Agora Papers\_, \_3\_(4). <https://doi.org/10.33262/ap.v3i4.1.123>

Ma, X (2020). A novel hybrid model by using convolutional neural network and long short-term memory for text sentiment analysis. \_Journal of Physics: Conference Series\_, \_1487\_(1), 012094. https://doi.org/10.1088/1742-6596/1487/1/012094