Detección de Noticias Falsas

José Antonio Rey Morales
Grado de Ingeniería Informática
Área de Inteligencia Artificial

Profesores
David Isern Alarcón
Friman Sánchez Castaño

TOC

Contexto y justificación

La creciente presencia de noticias falsas en internet, especialmente en redes sociales, plantea un problema significativo para la opinión pública y las instituciones. Las noticias falsas, a menudo diseñadas para generar reacciones emocionales, se propagan más rápido que las veraces, exacerbando tensiones durante eventos críticos como elecciones o desastres naturales.

Impactos recientes:

- Elecciones de EE. UU. 2020: La desinformación sobre fraude electoral generó discordia social.
- Mercados financieros: Falsos rumores afectaron empresas como Tesla y Facebook, provocando caídas bursátiles de hasta un 7%.
- Pandemia COVID-19: Desinformación sobre tratamientos impactó farmacéuticas como Moderna.

Objetivo general

El objetivo general es desarrollar un modelo de machine learning fiable que permita detectar noticias falsas, explorando y comparando diversas técnicas de Procesamiento de Lenguaje Natural (NLP) para clasificar noticias como verídicas o falsas.

Objetivos específicos

1. Representación textual básica

   Implementar técnicas como TF-IDF y Bag of Words y evaluar su desempeño.

2. Captura de contexto semántico

   Utilizar Word2Vec y N-gramas para captar relaciones semánticas.

3. Modelos basados en Transformers

   Implementar un modelo de tipo BERT.

Objetivos específicos

4. Optimización de modelos

   Ajustar hiperparámetros y aplicar técnicas de regularización y reducción de dimensionalidad.

5. Validación y generalización

   Validar los modelos en datos no vistos y generamos un snippet de código que cargue modelos y valide datos en tiempo real.

6. Conclusiones y lecciones aprendidas

   Obtenemos conclusiones del trabajo realizado y presentamos las lecciones aprendidas durante el trabajo.

Entorno de trabajo

Lenguaje y Entornos

- Lenguaje: Python 3.12.
- IDE: Visual Studio Code para desarrollo local.
- Computación intensiva: Google Colab para tareas de alto rendimiento.

Manipulación de Datos:

- NumPy: Operaciones matemáticas y manejo de arrays.
- Pandas: Manejo de datos tabulares en DataFrames.

Procesamiento de Lenguaje Natural (NLP):

- NLTK: Tokenización y eliminación de stopwords.
- spaCy: Lematización y análisis sintáctico con en_core_web_sm.

Entorno de trabajo

Visualización:

- Seaborn y Matplotlib: Generación de gráficos.

Modelos Avanzados de NLP:

- Gensim: Generación de word embeddings con Word2Vec (CBOW, Skip-gram).
- Transformers (Hugging Face):
  - DistilBertForSequenceClassification: Modelo de clasificación basado en BERT.
  - BertTokenizerFast: Tokenización eficiente para BERT.
  - Trainer y TrainingArguments: Gestión simplificada del entrenamiento.

Análisis y Otros:

- Collections.Counter: Análisis estadístico de tokens.
- Datasets: Carga y manejo eficiente de conjuntos de datos para modelos NLP.

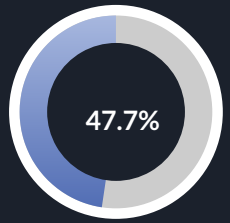
Análisis exploratorio

- Balance de clases: Bien balanceado, con una distribución de 47.7% noticias falsas y 52.3% noticias verdaderas.

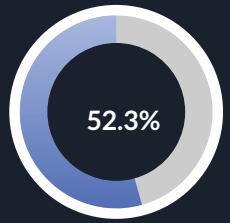
- Palabras más comunes antes del preprocesamiento: Predominio de stopwords como "the" y duplicados por capitalización, lo que justificó la limpieza textual.

Observaciones

Carácteres de media



Noticias falsas



Noticias verídicas

Preprocesamiento

El propósito general del procesamiento en trabajos de PLN es la de reducir el ruido del texto y así mejorar la calidad de los datos con los que se entrenarán los modelos.

- Conversión a minúsculas
- Eliminación de caracteres especiales
- Eliminación de stopwords
- Lematización
  - Ejemplo:
  - Antes: ['Trump', 'administration', 'getting', 'pummeled']
  - Después: ['trump', 'administration', 'get', 'pummel'].

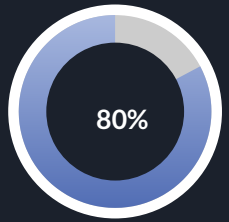
TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) es una técnica que mide la importancia de una palabra en un documento respecto al corpus completo, penalizando términos comunes.

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(t,d) : t \in d\}}$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

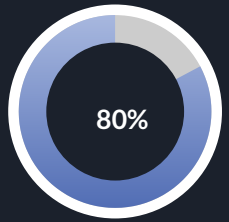


Split
entrenamiento/testing

max_features	accuracy	f1-score
10	0.80857	0.80816
20	0.83719	0.83686

BoW

BoW (Bag of Words) es una técnica de procesado de lenguaje natural que representa el contenido de los documentos sin tener en cuenta el orden (contexto). En un modelo BoW, se vectorizan las palabras en todo el corpus y se anota su frecuencia.

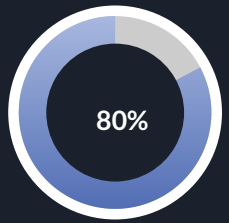


Split
entrenamiento/testing

max_features	accuracy	f1-score
10	0.79911	0.79798
20	0.82461	0.82376

N-Grams

N-Grams es una técnica de procesado de lenguaje natural que agrupa las palabras en n-gramas, por ejemplo bigramas (2-grams) serían grupos de dos palabras, trigramas (3-grams) serían grupos de tres palabras, esto ayuda a capturar el contexto del texto.

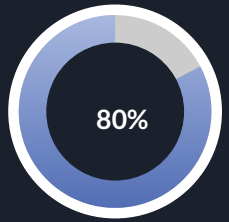


Split entrenamiento/testing

hiperparámetros	accuracy	f1-score
(1,2), max_features=10	0.8086	0.8082
(1,3), max_features=10	0.8086	0.8082
(2,3), max_features=20	0.7160	0.7035
(2,3), max_features=50	0.8778	0.8771
(1,2), max_features=50	0.9556	0.9554

Word2Vec

Word2Vec es una técnica de representación vectorial diseñada para capturar relaciones semánticas entre palabras. Genera vectores densos basados en similitud matemática, lo que permite identificar contextos y sinónimos más allá de la frecuencia de palabras.

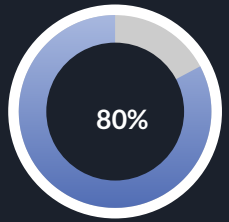


Split
entrenamiento/testing

hiperparámetros	accuracy	f1-score
Skip-Gram	0.9734	0.9733
CBOW	0.9725	0.9724

DistilBERT

BERT es un modelo de lenguaje avanzado basado en Transformers que utiliza atención bidireccional para entender el contexto de una oración en ambas direcciones. Esto lo hace particularmente eficaz en tareas como clasificación de textos, pero es un modelo complejo, difícil de parametrizar, medir, analizar y modificar.



Split
entrenamiento/testing

epoch	Training loss	Validation loss	accuracy	f1-score
1	0	0.0056	99.94%	99.95%
2	0	0.0036	99.96%	99.96%
3	0	0.0045	99.96%	99.96%

Primeras conclusiones

De esta primera generación de modelo podemos ver que TF-IDF supera a BoW, pero aumentar el número de características es más significativo que la técnica de modelado. En captura de contexto W2V tiene resultados similares con ambas técnicas y vemos que en N-grams la captura de contexto añade ruido mientras que aumentar las características mejora la precisión de manera muy significativa.

El modelo basado en transformers da señales de sobreajuste desde la primera epoch y no parece una bueno opción para continuar con el trabajo.

	Accuracy	Precision	Recall	F1-score
TF-IDF LR 10	0.80857	0.80810	0.80824	0.80857
TF-IDF LR 20	**0.83719**	**0.83676**	**0.83698**	**0.83686**
BoW LR 10	0.79910	0.79970	0.79736	0.79798
BoW LR 20	0.82461	0.82506	0.82317	0.82376
N-grams LR 1,2 10	0.80857	0.80810	0.80824	0.80816
N-grams LR 1,3 10	0.80857	0.80810	0.80824	0.80816
N-grams LR 1,2 20	0.83719	0.83676	0.83698	0.83686
N-grams LR 2,3 20	0.71603	0.73852	0.70769	0.70351
N-grams LR 2,3 50	0.87783	0.87930	0.87625	0.87714
N-grams LR 1,2 50	0.95556	0.95626	0.95485	0.95540
W2V LR sg1	**0.97338**	**0.97318**	**0.97351**	**0.97333**
W2V LR sg0	0.97249	0.97233	0.97255	0.97244
BERT	**0.99955**	**0.99915**	**1.00000**	**0.99957**

Hiperparametrización

Se implementó una hiperparametrización utilizando las técnicas vistas (TF-IDF, BoW, W2V) utilizando varios modelos con el objetivo de seleccionar un modelo con mejores resultados. N-grams se utilizó cómo parametrización, y DistilBERT se descartó en la hiperparametrización debido a sus resultados durante el entrenamiento.

También aplicamos PCA cómo técnica de reducción de dimensionalidad.

Hiperparámetros TF-IDF/BoW

- Max_features
- Frecuencia mínima
- Frecuencia máxima
- Rango de n-gramas

Hiperparámetros W2V

- Tamaño del vector
- Ventana de contexto
- Frecuencia mínima
- Algoritmo

Clasificadores

- Logistic regression
- Support Vector Machine
- Random Forest
- Passive-Aggressive Classifier

Análisis

Presentamos los mejores modelos generados, con su accuracy contra datos no vistos.

El mejor modelo ha sido Bag of Words con el clasificador Passive-Aggressive Classifier, con una accuracy general del 64,40%.

Modelo	accuracy
TF-IDF Random Forest	62,33 %
BoW PAC	64,40 %
W2V PAC	62,78 %
DistilBert	53,00 %

Reflexiones y lecciones aprendidas

Del trabajo realizado obtenemos las siguientes reflexiones

Contexto limitado: Los modelos que hemos generados analizan únicamente el cuerpo de la noticia, por lo que perdemos información relevante que no está incluída en el texto.

Cómo mejorar la precisión del modelo:

- Modelar el entorno sociocultural
- Análisis de sentimiento
- Verificar la veracidad de lugares y personas
- Análisis de noticias similares
- Web scrapping constante