UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

Estudio Comparativo entre Modelos Basados en la Arquitectura Transformer para Detectar Texto Maligno

Autor: Ponce Bravo, Bryan Steeven
Tutor: Ing. Benavides Astudillo, Diego Eduardo, Mgtr.

Tabla de contenidos

01

Introducción

Introducción

La ingeniería social es un tipo de ataque cibernético en el que el atacante engaña y estafa a la víctima utilizando manipulación psicológica, abusando de su comportamiento desprevenido o sus rasgos ingenuos de la personalidad.

El phishing es un delito donde se emplea ingeniería social y técnicas de engaño para robar los datos de identidad y las credenciales de cuentas financieras de los consumidores. El phishing mediante uso de sitios web fraudulentos, frecuentemente son una copia de la página real, es una de las formas por las que se puede llevar a cabo este tipo de ataque de ingeniería social.

Estado de arte

Tabla I: Estado del arte de investigaciones relacionadas en la detección de phishing con DL y Transformer

Artículo	Email	Página web	NLP	Precisión	Aplicación
[9]	No	URL	Sí	N/A	N/A
[10]	Sí	N/A	Sí	87.8%	Sí
[11]	Sí	N/A	Sí	99%	N/A
[12]	No	URL	Sí	96.66%	N/A
[13]	No	URL	Sí	95.7%	N/A
[14]	Sí	N/A	Sí	N/A	N/A
[15]	No	URL	Sí	86.80%	N/A
[16]	Sí	N/A	Sí	98.66%	N/A
[17]	Sí	N/A	Sí	98.67%	N/A
[18]	Sí	N/A	Sí	99.68%	N/A
Nuestro estudio	No	Texto	Sí	93.32%	Sí

Objetivo general

Analizar y comparar diferentes modelos basados en la arquitectura Transformer junto con el procesamiento de lenguaje natural para detectar ataques de phishing sobre el texto contenido en esos ataques.

Objetivos específicos

- Determinar qué aplicaciones usan actualmente las tecnologías Transformer para detectar ataques de phishing sobre el texto contenido en esos ataques.
- Determinar cuál de los modelos Transformer brinda mejores resultados para la detección de ataques de phishing sobre el texto contenido en esos ataques.
- Entrenar un modelo con el algoritmo Transformer y ponerlo en producción para la detección de ataques de phishing.
- Ofrecer un aplicativo en un navegador web para detectar ataques de phishing que sea amigable para el usuario.

02

Marco conceptual

Marco conceptual

Phishing

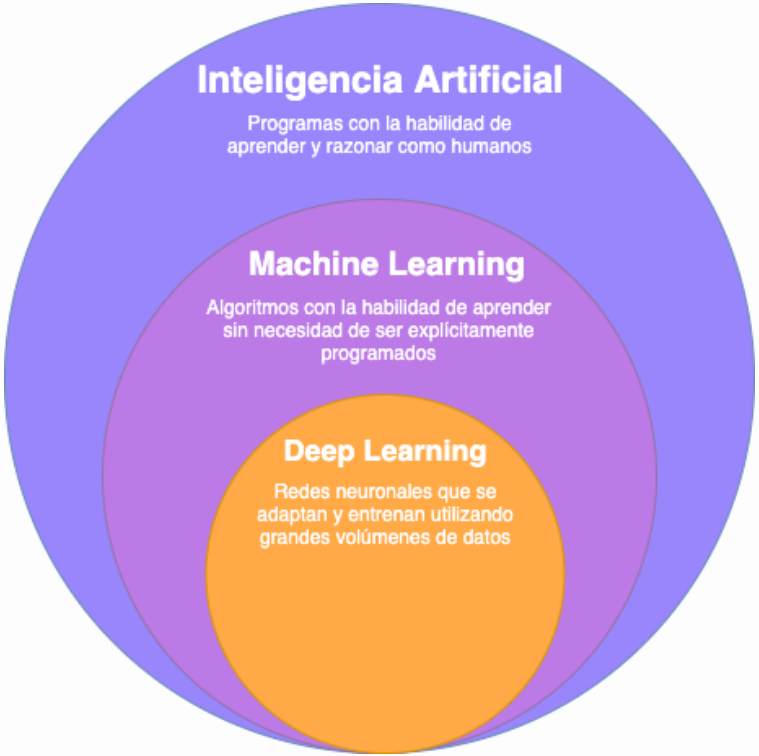
Es una práctica maliciosa generalmente realizada a través de correo electrónico o páginas web, con el objetivo de robar información personal.

Existen varios tipos de ataques de phishing, tales como: phishing Tradicional, spear phishing, whaling, smishing, vishing, clone phishing, business email compromise, angler phishing y **web phishing.**

NLP

Es un campo que permite a los humanos comunicarse y expresarse a través del lenguaje, en este contexto se basa en la asociación de signos con significados concretos.

Marco conceptual



Marco conceptual

Deep Learning

Es un subcampo de la AI que se centra en entrenar RNN para aprender y realizar tareas de procesamiento de datos. Las capas de neuronas artificiales de RNN procesan datos de forma jerárquica y progresiva. DL desempeña un papel importante en una amplia gama de aplicaciones, incluido el procesamiento de imágenes, NLP, visión artificial y reconocimiento de voz.

Modelos Transformer

Los modelos Transformer pueden entrenarse en grandes cantidades de datos y se adapta a tareas específicas. Entre los modelos que utiliza la arquitectura Transformer se encuentran BERT, GPT, T5 y XLNet, los cuales son objeto de este estudio.

Marco conceptual

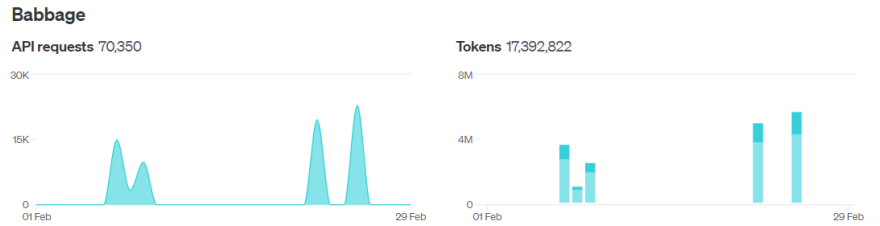
Modelos Transformer

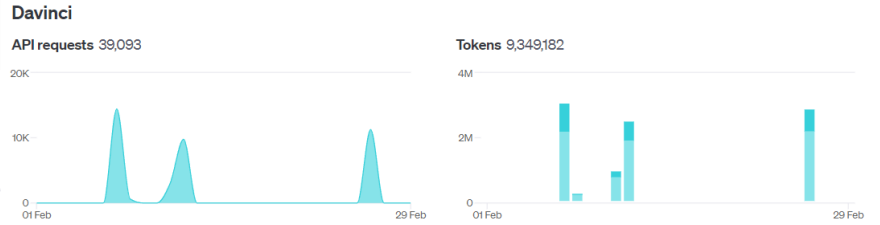
Tabla IV: Comparativa de Rendimiento de Modelos Transformer

Modelo	Costo de Acceso	Costo de Infraestructura	Costo de Operación	Observaciones
GPT	Pago	Alto	Moderado	Requiere GPUs
BERT	Gratis	Moderado	Bajo	Optimizable
T5	Gratis	Alto	Moderado	Alta precisión
XLNet	Gratis	Alto	Moderado	Buen rendimiento en tareas específicas

Costos de GPT

Modelo	Uso
davinci-002	$2.00 / 1M tokens
babbage-002	$0.40 / 1M tokens

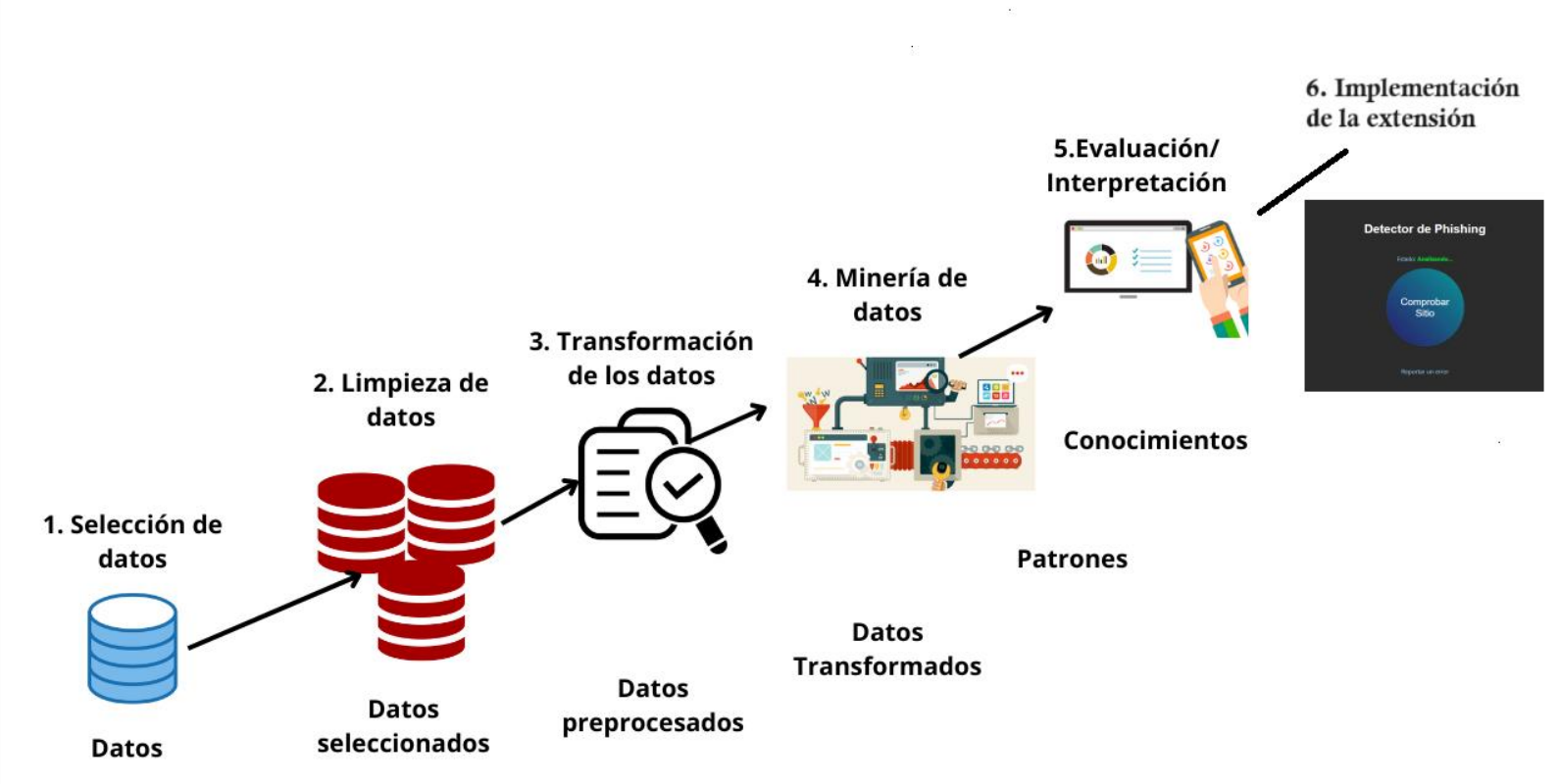




03

Metodología

KDD


1. Selección de datos
Datos
2. Limpieza de datos
Datos seleccionados
3. Transformación de los datos
Datos preprocesados
4. Minería de datos
Datos Transformados
Patrones
5.Evaluación/ Interpretación
Conocimientos
6. Implementación de la extensión
Detector de Phishing
Comprobar Sitio

Fase de selección

Selección del conjunto de datos

- **Investigaciones**: Scopus, Scielo, Redalyc.org, IEEE y WebOfScience.
- **Selección**: Conjunto de datos de libre acceso Phishload
- **Descomprimir**: Archivo de tipo SQL, abrimos en HeidySQL y exportamos a un formato CSV
- **Tabla websites**: La columna context, que contiene el código HTML de todas las páginas web, y la columna phishing que indica si una página web es phishing o ham

Fase de Transformación

Transformación

- **Tokenización** — Vectorizar cada palabra de un texto anterior, creando una secuencia de palabras con significado
- **Encoding** — Codificar las palabras de una cadena de texto, que asigna un número entero a cada palabra a medida que aparece.
- **Padding** — Longitud máxima maxlen que debe tener la cadena introducida en el ´ algoritmo DL
- **Representación de características con Keras Embedding y GloV** — Funciona mapeando cada palabra del flujo de entrada a una representación vectorial

Fase de Minería de datos

Ejecución de algoritmos Transformer

Se implementaron y entrenaron cuatros modelos Transformer (GPT, T5, BERT y XLnet)

Selección del modelo

Se seleccionó BERT como el modelo más adecuado para la aplicación

Implementación

Los modelos Transformer, se lo desarrollaron con el lenguaje Python

Trabajo adicional Comparación de modelos

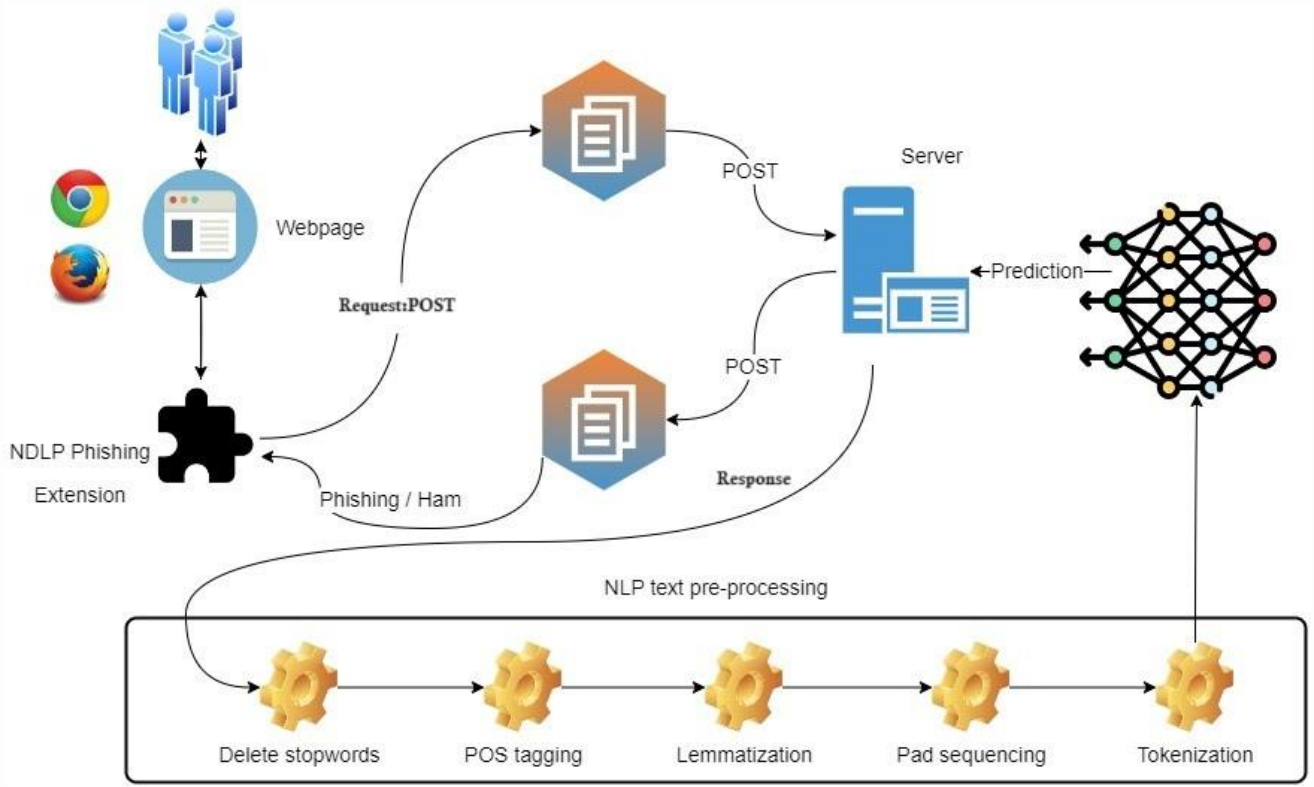
GPT tiene dos modelos entre ellos davinci y babbage que son para clasificar texto.

Modelos GPT

Tabla VIII: Acurracy de los modelos de GPT babbage y davinci comparados

Modelo	Acurracy	Tiempo de ejecución
davinci-002	51.35 %	90 minutos
babbage-002	80.76%	60 minutos

Funcionamiento de la extensión



04

Resultados

Métricas de los modelos

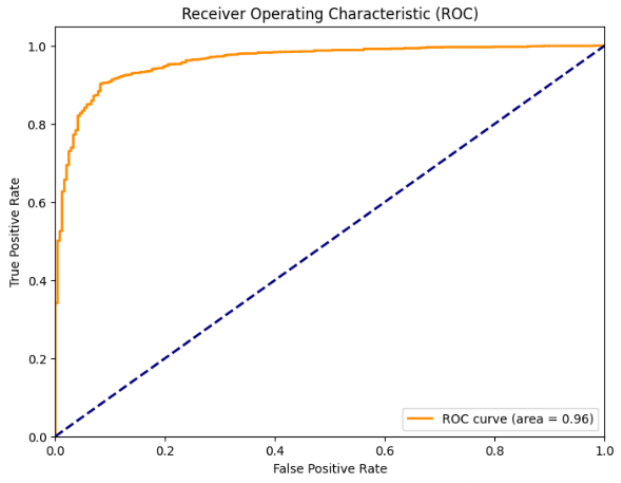
Tabla VI: loss, acurracy, val_loss y val_accuracy con L = 200 y con 5 epochs

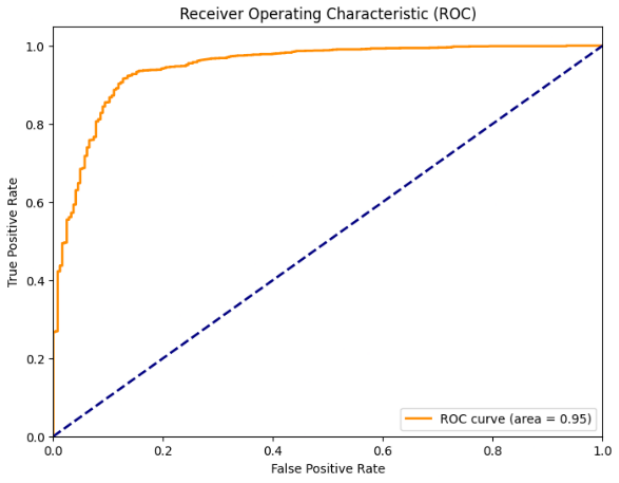
Algoritmo	L	epoch	loss	accuracy	val-loss	val-accuracy	Tiempo de ejecución
BERT	200	5	0.08	0.97	0.19	0.93	1 hora y 19 minutos
GPT	200	N/A	N/A	0.81	N/A	0.81	1 hora
T5	200	5	0.19	0.92	0.18	0.92	2 horas 68 minutos
XLNet	200	5	0.16	0.93	0.19	0.92	10 horas 30 minutos

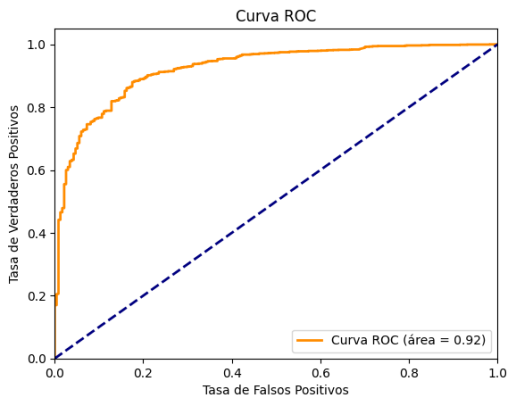
Tabla VII: Métricas de los modelos Transformer

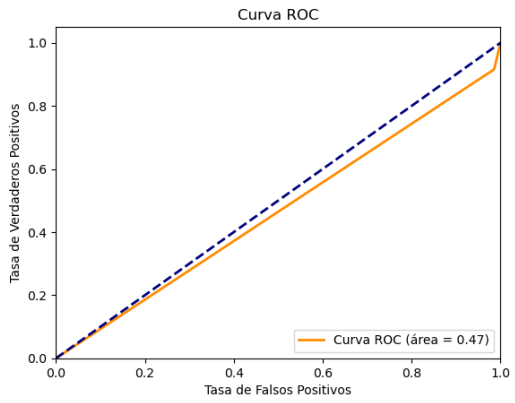
Algoritmo	Precisión	Recall	F1 score
BERT	0.96	0.95	0.96
GPT	0.86	0.91	0.89
T5	0.94	0.96	0.95
XLNet	0.95	0.97	0.96

Curva de los modelos BERT, XLNet, T5 y GPT









06

Conclusiones y recomendaciones

Conclusiones

En este trabajo se realizó un análisis comparativo de los modelos basados en la arquitectura Transformer GPT, T5, BERT y XLNet, para la detección de páginas web de phishing. Según nuestro estudio, el modelo BERT obtuvo un acurracy del 93.34%, en un segundo lugar XLnet obtuvo un 92.88%, T5 obtuvo un acurracy 92.58% y como último lugar GPT obtuvo un acurracy 80.76%.

Un aporte significativo de nuestro estudio en comparación con otros trabajos del estado del arte analizados, es la capacidad para detectar phishing mediante el análisis del contenido textual de las páginas web. Para esto, primeramente se hizo un pre procesamiento mediante NLP al texto HTML, resultando en un texto limpio sin código HTML. Luego se usa la tecnica de word embedding con GloVe para incrustar de palabras, y aprovechar más las conexiones semánticas entre ellas.

Recomendaciones

Debido a la complejidad y la demanda de recursos de los modelos Transformer, la recomendación principal es disponer de una computadora altamente equipada.

Limitar el tamaño de los conjuntos de datos a un máximo de 200 palabras por entrada para optimizar el uso de recursos y tiempo en el entrenamiento y ejecución de modelos Transformer, especialmente en tareas de detección de phishing. Esta técnica permite un procesamiento más eficiente sin comprometer significativamente la efectividad o la precisión de la detección. Se puede mantener un equilibrio entre el rendimiento y los recursos enfocados en segmentos de texto más concisos, lo que facilita una implementación más ágil y sostenible.

Se recomienda la utilización de Python 3.10.2 para facilitar el desarrollo efectivo de algoritmos utilizando modelos Transformer de DL. La compatibilidad comprobada con TensorFlow, que es esencial para las operaciones de clasificación de texto, respalda esta recomendación. Durante el transcurso de este proyecto, se descubrió que otras versiones de Python tenían incompatibilidades significativas con TensorFlow.

Gracias