Analisis de contenido de texto basado en procesamiento de lenguaje natural con BERT

Jairo Jesus Gonzalez Hernandez
Esudiante Ing. sistemas
Universidad del Norte
Barranquilla - Colombia
jairojg@uninorte.edu.co

Jesus Daniel Angulo Rivera
Estudiante Ing. sistemas
Universidad del Norte
Barranquilla - Colombia
angulojg@uninorte.edu.co

Andres Felipe Meza Caballero
Estudiante Ing. sistemas
Universidad del Norte
Barranquilla - Colombia
amezaf@uninorte.edu.co

Katherine Sofia Palacios Salgar
Tutora de proyecto
Universidad del Norte
Barranquilla – Colombia
kpalacio@uninorte.edu.co

Wilson Nieto Bernal
Tutor de proyecto
Universidad del Norte
Barranquilla – Colombia
wnieto@uninorte.edu.co

***Abstract* — This article focuses on trying to alleviate problems related to content analysis. We will discuss the use of different models for classification in Machine learning. We take this approach to solve some problems related to qualitative analysis, such as reliability over time and the decline of skilled labor. We do this to automate a process that usually requires considerable amounts of time and resources, such as trained humans and long lead times. We explored the use of different techniques like Random Forest and K-Nearest Neighbor, we also tried different bag of words methods to encode the text. We also evaluated a prototype of the proposed solution with Bidirectional Encoding Representations of Transformers (BERT) under a dataset for detection of fake news due to scope limitations, However, it is applicable to another corpus and other text context. Finally, with AWS services we will implement a system for the creation of an API that can be used by the common user and implemented in their classification systems.**
Keywords—NLP, text classification, BERT, content analysis.

I. Introduction

En los últimos años hemos visto un gran avance en las técnicas y modelos relacionados al procesamiento de lenguaje natural, en su mayoría luego del trabajo de Mikolev, 2013 [1]. Dichos avances nos han permitido automatizar procesos como la atención de usuarios por medio de chatbots y rediseño de procesos de negocios como lo menciona Mustansir, 2022 [2].

En este proyecto, nos enfocaremos en la creación de una herramienta que nos permita entrenar modelos para análisis de contenido basados en BERT. También mencionaremos otras pruebas que se realizaron con otros modelos y porqué se hizo la decisión de seguir con BERT, aunque hubo modelos más atractivos en ciertos momentos.

El objetivo final de este proyecto será la creación de diferentes modelos para varios contextos donde se requiera automatizar el análisis de contenido de un texto cualquiera y el despliegue de dichos modelos en una aplicación web que nos permita hacer análisis con estos modelos de una manera amable para usuarios finales.

II. Problema

El análisis de contenido es una tarea complicada ya que usualmente se necesita de la participación de expertos. Dicho proceso toma tiempo y un gran costo porque requiere contratar a personal especializado en el área de estudio para poder evaluar el texto y extraer los componentes o realizar la categorización deseada.

El procesamiento de lenguaje natural (NLP) permite aplicar técnicas de Machine Learning (ML) al lenguaje humano. Esta posibilidad permite hacer análisis para los cuales antes se consideraba necesaria la intervención humana y el juicio de expertos, entre esas tareas encontramos el análisis de contenido. El uso de computadores nos ahorra una gran parte de los costos asociados a la contratación de personal especializadd, permite mejorar el tiempo necesario para dicha actividad y permitirá tener un criterio más estandarizado para la realización de dicha tarea.

El problema que queremos resolver es la creación de una herramienta que nos permita la creación de dichos modelos. De modo que se pueda automatizar el proceso de análisis de contenido.

III. Justificación

La importancia de este proyecto radica en la automatización de un proceso que normalmente tiende a ser demandante en términos de tiempo y recursos. El análisis de contenido incluye la clasificación de texto, como por ejemplo el clasificar el contenido de un periódico en base a sus temáticas. Este proceso tiene varias complicaciones que se podrían solucionar con el uso de modelos de NLP. La solución de dicho problema tiene aplicaciones tanto en investigación como fue el caso original como para la industria, esto ya que reduce costes, ahorra tiempos y permite mantener un mismo estándar para la toma de decisiones a lo largo del tiempo.

IV. OBJETIVOS

Objetivo General

Modelar y diseñar de una herramienta que nos permita crear modelos basados en procesamiento de lenguaje natural con el fin de desarrollar análisis de contenido.

Objetivos específicos

1. Identificar los componentes claves para el modelado y diseño de una herramienta que nos permita crear modelos basados en procesamiento de lenguaje natural con el fin de desarrollar análisis de contenido.
2. Diseñar la arquitectura lógica de la solución (Infraestructura, Datos, App, Procesos y UI) asociada con aplicación basada en procesamiento de lenguaje natural.
3. Desarrollar el prototipo de aplicación basada en procesamiento de lenguaje natural, articulado con la arquitectura lógica de la solución
4. Validar el prototipo de aplicación basada en procesamiento de lenguaje natural, a través de un instrumento de calidad estandarizado.

-

MARCO TEORICO

Encontramos una amplia variedad de trabajos relacionados a el uso de modelos de NLP aplicados a la categorización de noticias (como los veremos en las siguientes menciones) y texto en general usando modelos como Bidirectional Encoder Representation for Transformers (BERT) y Transformers, así como el impacto de soluciones de tecnología.

De acuerdo con Ping en [3] se presenta que el avance de la tecnología ha provocado un aumento en la circulación de la información. Es por eso, que tiene como objetivo un analizador de sentimientos en la información que se encuentra circulando. Este artículo usa los modelos BERT y redes neuronales (CNN) para hacer dicho análisis.

Por otra parte, Kuncahyo en 2021 [4], tiene como objetivo el impacto del procesamiento Big Data en las tareas NLP basado en Deep Learning. Esto nos provee de un contexto y fundamento para hacer nuestros modelos.

En [5] se presenta un modelo en BERT para la detección de noticias falsas. Es un método que construye un texto con patrón a nivel lingüístico para integrar la reivindicación y las características de manera pertinente. Este es un ejemplo interesante del uso de BERT para clasificación de noticias en el mundo real y en un contexto social.

En [6] se presenta una comparación de modelos tradicionales de clasificación de texto como lo serian Support Vector Machine y Naive Bayes contra modelos algo más recientes como BERT y XLNet contra un dataset de 50,000 reseñas en inglés, dejando como ganador a XLNet con un 8\% más de accuracy que el modelo clásico con mejor desempeño. Esto nos ayuda a ver como se realizan este tipo de comparaciones entre modelos basados en clasificadores lineales contra modelos que se basan en redes neuronales.

Asimismo, en [7] evidencia como el rápido desarrollo de tecnologías de Machine Learning y NLP han provocado que las tareas de clasificación de texto sean cada vez menos manuales y más automatizadas debido a las mejores que se han obtenido en los últimos años. También se propone en el estudio el uso de un modelo híbrido para mejorar estos resultados. Esto resulta importante para este proyecto ya que la motivación e inicios de este proyecto partió de la misma premisa; el automatizar la clasificación de noticias hecha por un grupo de personas en 6 meses.

Comparativamente, en [8] se intenta analizar diferentes modos de hacer finetunning a un modelo de BERT, desde agregar nuevas capas convolucionales, BiLSTM o métodos que describen en su artículo. Esto debido a que los resultados de hacer un finnetunning no siempre son óptimos y es un problema con el que el investigador se enfrenta a lo largo del desarrollo de estos modelos y son una aproximación teórica a lo que se estaba desarrollando.

Por otra parte, [9] tiene como problemática la clasificación de información redundante que interfiere con la precisión. Es por ello por lo que crearon un modelo de clasificación de etiquetas de categorías BERT-DBLCA.

Así mismo se puede ver como en [10], se evidencia como algunos modelos clásicos aún tienen un desempeño bastante aceptable en el mundo moderno con un recall de 92.87% para un modelo Naive Bayes clasificando automáticamente noticias de NBC. Este artículo apoya la idea de que no siempre es necesario usar modelos que sean el estado del arte para solucionar problemas complejos, si no que aún tienen su espacio y nos pueden dar buenos resultados si estamos dispuestos a sacrificar algo de desempeño a cambio de mejorar los tiempos en los que se obtienen la clasificación.

Finalmente, en [11] se presenta como las redes sociales y las noticias falsas tiene una relevancia importante en el número creciente de casos de discursos de odio hacia los migrantes y refugiados. En este estudio, se usaron tres métodos; una encuesta que mide las opiniones de los lugareños, análisis de redes sociales (SNA) y una encuesta experimental. Esto provee un ejemplo de análisis de noticias en un contexto de migración y como estas noticias impactan la percepción de los locales

V. METODOLOGÍA

Inicialmente, para el diseño y desarrollo de la herramienta se planteó la necesidad de su creación dentro de una investigación sobre migración donde necesitaban

automatizar o al menos ayudar a la categorización de noticias con base en ciertos contenidos que se encontraba en dicho texto. El trabajo manual tomo alrededor de 6 meses con personas y esto motivo la propuesta de este proyecto.

Los primeros datos con los que se trabajó vinieron de la investigación anteriormente mencionada y constaban de una categorización por párrafos de noticias de diferentes periódicos. Dicha categorización fue hecha manualmente durante un mes por 2 personas. Posteriormente se trabajaría con una versión por oraciones de este dataset con las mismas categorías, pero con diferentes tamaños de muestras por categoría, dicha característica hizo que se tuviera que replantear un par de aspectos como el método de entramiento y la manera en la que generaríamos las predicciones. No es viable socializar mucha información sobre dicha investigación por lo que se intentará hacer referencia a aspectos generales y no específicos. Nuestro dataset de entrenamiento constaba de 5 categorías con 100 muestras de cada una, en total 500 secuencias de texto.

Los primeros modelos eran implementaciones de Random Forest que trabajaban a partir de aplicar One Hot Encoding sobre los datos de entrenamiento. Sobre estos datos aplicábamos procedimientos de modos que se eliminaran palabras basura (Stopwords) y se redujera el número de palabras. El modelo en un inicio funcionaba relativamente bien, en términos de accuracy y tiempos de ejecución, para lo necesitado en su momento, pero debido a la cercanía de 2 categorías se vio necesario el incluir más información para evitar confusión entre dichas categorías. Por lo que se agregó entonces una dimensión de Sentiment Analysis y extraer la aparición de términos que fueron considerados clave por los expertos en la clasificación.

La implementación de dichos cambios no fue exitosa como se esperaba y no se logró superar un accuracy de 60%. Se optó por descartar la adición de estos componentes ya que sacrificaban mucho la independencia con respecto al juicio de expertos, debido a que requeriría la identificación de términos clave y el análisis de sentimientos era algo que tenía sentido en las categorías que estábamos buscando clasificar pero que no necesariamente se podía generalizar.

El flujo anteriormente mencionado se intenta aplicar a otros clasificadores lineales como lo serian KNN, Naive Bayes y Support Vector Machine, pero el alcance del proyecto solo permitió implementar el primero. Los resultados de KNN en términos de métricas no cumplieron con las expectativas, no superando un accuracy de 30% en el mejor de los casos por lo que se exploraron más modelos en la literatura dando como resultado la incorporación de IBM Watson, spaCy y Hugging Face con sus múltiples modelos, entre ellos BERT.

En esta instancia, la participación de un experto en lingüística para que proveyera feedback al proceso se hizo necesaria para verificar los resultados y avances. Las recomendaciones del experto nos llevaron a trabajar con el segundo dataset anteriormente mencionado el cual estaba clasificado por oraciones en lugar de párrafos.

A partir de esta etapa se evidencia la oportunidad de crear una herramienta más genérica que no fuera demasiada específica para las necesidades del proyecto de investigación si no que estuviera medianamente parametrizada de modo que se pudiera configurar para otras clasificaciones.

En un inicio se intenta trabajar de igual manera que como se hace con los modelos anteriores, dicha tarea fue difícil de ejecutar debido al alcance de este proyecto. El hecho que el dataset estuviera desbalanceado hace que fuera necesario descartar datos que serían valiosos ya que de por si estábamos trabajando con un numero de muestra relativamente bajo.

A partir de aquí la reformulación del problema fue necesaria. Inicialmente, con los clasificadores lineales, se creaba un solo modelo que para una secuencia de texto asignara una categoría. Esto requería el garantizar la uniformidad en los datos de entrenamiento. Esto se intentó con BERT, pero los resultados no fueron los esperados. en términos de acuraccy, estaba por debajo de 60%. Se obto por cambiar la manera en que estaba codificado el segundo dataset y se permitió que una secuencia de texto, que este caso era una oración, pudiera ser asignada a varias categorías. Este cambio permitió aprovechar todas las muestras de una categoría sin que se viera afectada la clasificación de otra categoría. No obstante aumento los tiempos de entrenamiento pasando de un solo entrenamiento por dataset a un entrenamiento por categoría dentro del dataset y así mismo todo se multiplico por el número de categorías: el almacenamiento requerido, tiempo de corrida y pequeños detalles como llamar varios modelos en lugar de solo uno.

Una vez que se tuvieron estos cambios en mente se pudo proseguir. En esta ocasión se realizó un proceso conocido como finetunning, dicho proceso consiste en aprovechar el entrenamiento previo realizado sobre redes neuronales y sobre este hacer un entrenamiento posterior con un dataset más cercanos a las necesidades inmediatas de modo que quede un modelo más ajustado a las categorías que se tienen lo que debería desencadenar en mejores resultados.

Para realizar este proceso se escogió el modelo “bert-base-uncased” provisto por la plataforma Hugging Face. Sobre este se desarrolla un pipeline de modo que a partir de un par de configuraciones básicas se pudiera crear un modelo por cada categoría dentro del dataset y casi automáticamente se subiera a dicha plataforma. Esta plataforma nos proveyó de varias facilidades para el manejo de dichos modelos ya que permitía almacenarlos de manera remota y tenía un sistema de validación de acceso para evitar que personal externo accediera a nuestros modelos ya creados.

Con la generación de modelos en un estado aceptable y funcional procedimos a la siguiente etapa, siendo esta el despliegue de nuestro modelo en forma de API. Para esto se evaluaron alternativas como Azure y los servicios en la nube de Google pero se decidió usar AWS por familiaridad. Este proceso se detallará en la siguiente sección (Metodología de la analítica).

VI. METODOLOGIA DE LA ANALITICA

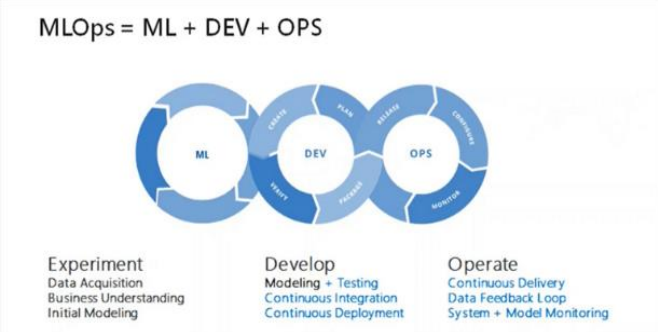

Ilustración 1: Ciclo de vida de MLOps tomado de Tech in 3

Según [5] el fin del ciclo MLOps es proveer un proceso de fin a fin para el proceso de diseñar, construir y manejar software que sea reproducible, comprobable y pueda evolucionar en el tiempo. Dicho proceso se logra por medio de mejoras constantes y su implementación con buenas prácticas.

Para el presente proyecto se decidió aplicar dicha metodología ya que una vez se desarolló la herramienta y se pudo generar los primeros modelos, esta metodología permitía pasar los modelos a un entorno de producción a medida que se generaban mejores modelos sin causar graves repercusiones o alterar grandemente el funcionamiento de la aplicación.

Pasos del ciclo de vida:

Planificación

Resulta importante, como para todo proyecto, el planificar y delimitar el alcance y objetivos del software. Así mismo es necesario decidirse por una arquitectura, patrones de diseño y evidenciar posibles problemas antes de que ocurran para estar preparados.

Datos

En esta fase se analizan los datos y fuentes de estos que ayuden a resolver el problema y la manera en que se extraerán (ETL, pipelines e insumo directo). Aquí también es necesario pasar por una etapa de Feature Engineering. Usualmente se pasa por las siguientes subetapas:

1. Análisis exploratorio
    a. Distribución de las variables
    b. Análisis de correlación.
    c. Análisis de integridad de los datos.
2. Transformaciones
    a. Identificar y tratar datos atípicos
    b. Seleccionar variables más relevantes
3. Preparación de datos

Modelado

Una vez que se haya las variables más importantes en la etapa anterior, se procede a identificar posibles modelos y métodos que nos ayuden a encontrar una solución satisfactoria.

- Selección de posibles modelos (En esto se centró nuestra sección de metodología).
- Métodos estadísticos que nos ayuden a tomar mejores decisiones.
- Optimización de hiperparametros.
- Prueba de rendimiento.

Pruebas y despliegue de la solución

Con los modelos ya puestos a punto, se puede desplegar la solución. Se debe tener en cuenta lo siguiente:

- Las dependencias deben ser satisfechas en el entorno de producción.
- Cambios en las métricas una vez que llega a un entorno de producción.
- Hallar métricas relevantes para la solución y no solo para el desempeño del modelo.
- Asegurar la conexión del modelo con la interfaz de usuario en caso de que exista.

Monitoreo

Es necesario hacer un monitoreo constante del modelo y su interacción con los nuevos datos. Validar que las predicciones y/o pronósticos. Identificar posibles problemas y plantear soluciones. Como se menciona en la etapa anterior es probable la ocurrencia de errores al pasar a un entorno de producción y el constante uso puede revelar errores o situaciones no especificadas. Así mismo, es importante verificar el correcto funcionamiento de la infraestructura de la solución.

VII. DISEÑO DE LA INVESTIGACIÓN

Durante el desarrollo de este proyecto se evaluaron varias alternativas como fue anteriormente mencionado, pero nos centraremos en la versión final desarrollada con HF.

Los datos con los que se trabajó inicialmente provenían de una fuente primaria, siendo noticias de periódicos colombianos y griegos. Pero al momento de generalizar dicha herramienta, el origen de los datos del proyecto era

indistinto en tanto que estuvieran en un formato csv o xlsx. Los datos del proyecto pasaban por un ciclo de preprocesado estándar para NLP que incluye, pero no está limitado a Tokenization, Stemization y remover signos de puntuación.

VIII. ARQUITECTURA LÓGICA DE LA SOLUCIÓN

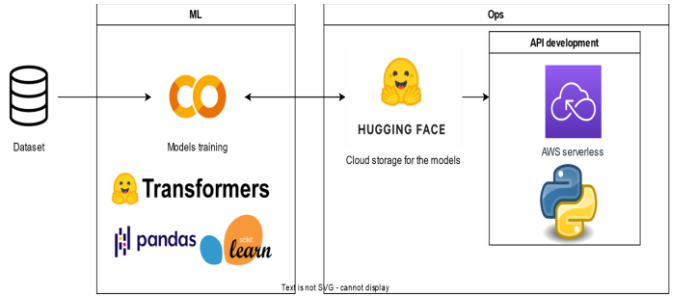

Ilustración 2: Arquitectura lógica de la solución de elaboración propia

La herramienta que fue desarrollada se implementó en un Jupyter Notebook alojado en Google Collab para hacer uso de su infraestructura de cómputo y haciendo uso de las librerías como se muestra en la ilustración 2. Los modelos creados son posteriormente subidos a la nube de almacenamiento de Hugging Face desde donde pueden ser accedidos por cualquiera con la librería Transformers. Ya con esto se puede acceder desde el api.

IX. ARQUITECTURA FÍSICA DE LA SOLUCIÓN

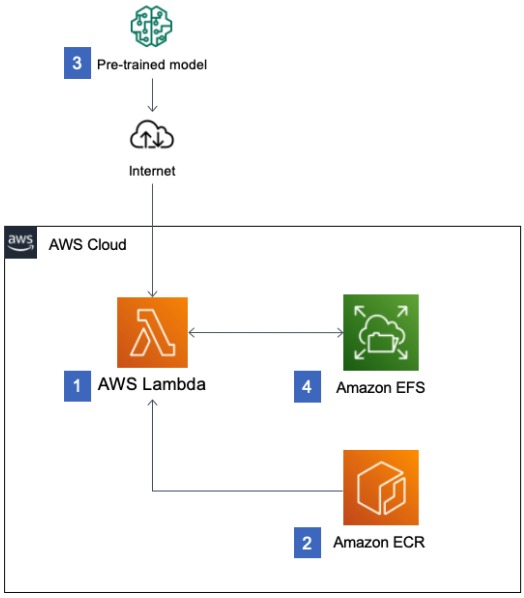

Ilustración 3: Arquitectura física tomada de AWS

Esta es la arquitectura que se tomo para la implementación de la API haciendo uso de la infraestructura de Amazon Web Services. Se aloja el modelo entrenado a AWS de modo que se pudiera usar desde la función Lambda y desde aquí se pueda acceder por los clientes.

X. PROTOTIPO

Para la evaluación de la herramienta se tomó un dataset para la detección de fake news y ese modelo fue el utilizado para hacer el ciclo de MLOps. Las imágenes (ver ilustración 9 a ilustración 14) a continuación son referentes a dicho proceso. El corpus con el que trabajamos (ver ilustración 4 a 8) marcaba las noticias como “REAL” o “FAKE” y fue una adaptación propia (de formato y no de contenido) para el corpus conocido como “ISOT Fake News de University of Victoria”.

Dicha transformación consto de concatenar verticalmente los datasets de las dos categorías dentro de uno solo, agregando una nueva columna llamada “label” que identificaba de dónde venía el registro al que pertenecía, si al corpus de noticias verdaderas o falsas.

Imagen del corpus:

	title	text	label
0	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL
...	...	...	...
6330	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE
6332	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
6334	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

6335 rows × 3 columns

Ilustración 4 Dataframe obtenido de los datos:

Para este ejercicio se tomaron 1400 registros para entrenar, de los que se apartaron 420 para evaluación de metricas.

Proyección de nuestro corpus en un espacio bidimensional, también conocido como Latent Semantic Analysis (LSA):

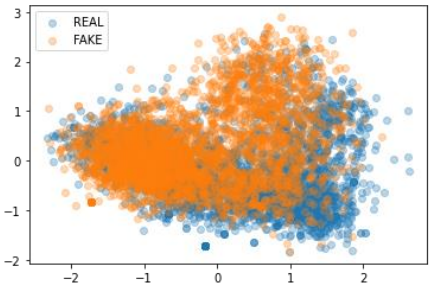

Ilustración 5: LSA del dataframe de creación propia

En la ilustración 5 podemos ver que hay categorías bastante solapadas, esto podría ser peligroso ya que implicaría que sería difícil lograr un buen clasificador. Se analizo la variabilidad explicada por los vectores resultantes de realizar Principal Component Analysis y se observó que para este dataset, no se estaba explicando más del 15% por lo que aun cabe la posibilidad que en una dimensión mayor

sea posible evidenciar una mejor diferenciación entre las categorías que buscamos separar.

Wordcloud de la totalidad del corpus sin remover stopwords o ninguna forma de limpieza sobre los datos:

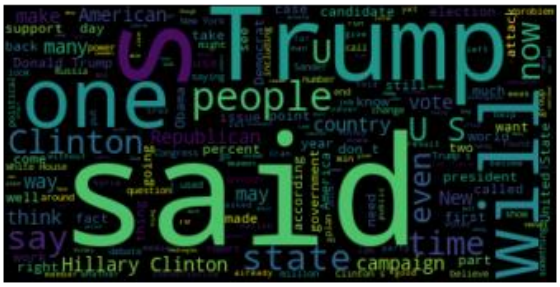

Ilustración 6: Wordcloud de la totalidad del corpus de elaboración propia

Wordcloud por categoría:

- Fake

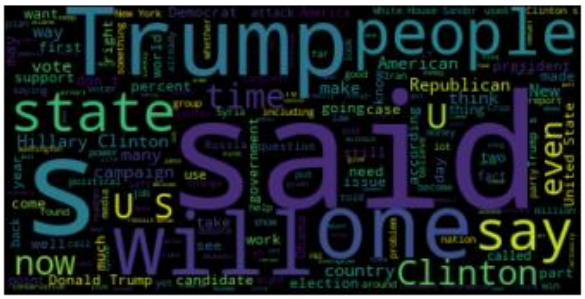

Ilustración 7: Wordcloud de la categoría Fake de elaboración propia

- True

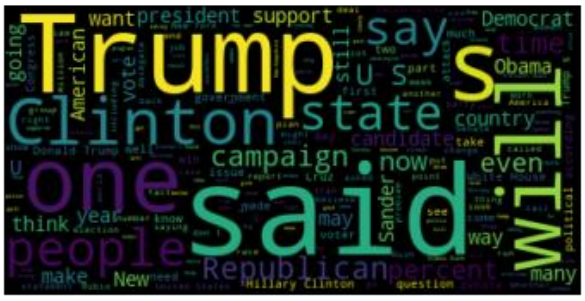

Ilustración 8: Wordcloud de la categoría True de elaboración propia

Librerías usadas:

```
from distutils.command.config import config
import sys
import argparse
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, recall_score, precision_score,precision_score, f1_score,confusion_matrix
import torch
from transformers import AutoConfig, TrainingArguments, Trainer
from transformers import BertTokenizer, BertForSequenceClassification
from transformers import EarlyStoppingCallback
from google.colab import drive
drive.mount('/content/drive')
```

Ilustración 9: Librerías usadas en la implementación

Configuraciones iniciales de la herramienta:

```
##  LOAD FILE
# input dataset in .csv format
INPUT_PATH = "/content/drive/MyDrive/PF/training_corrected.csv"
# column name with sequences to analyze
COLUMN_NAME_TO_READ = "text"

##  PARAMETERS
# number of posible categories por column
NUM_LABELS = 2
# sample size to train the detection of a categorie
MAX_SAMPLES = 2000
# categories to train
BLAMES = [
    "true",
    "fake"
    ]
```

Ilustración 10: parámetros de la herramienta

Modelos alojados en HuggingFace.co:

Ilustración 11: Captura de los modelos alojados en la nube

Archivos del modelo para detección de noticias falsas:

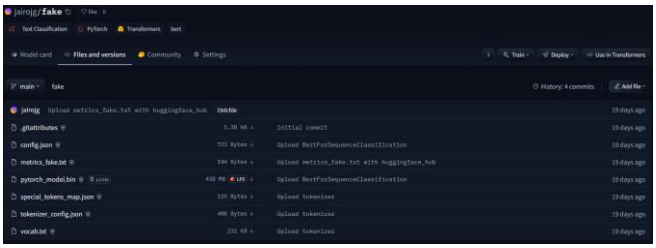

Ilustración 12: Archivos del modelo "Fake" alojados en la nube

Evaluación de la Api haciendo uso de Postman:

- Para una noticia que era considerada “Real” en nuestro dataset:

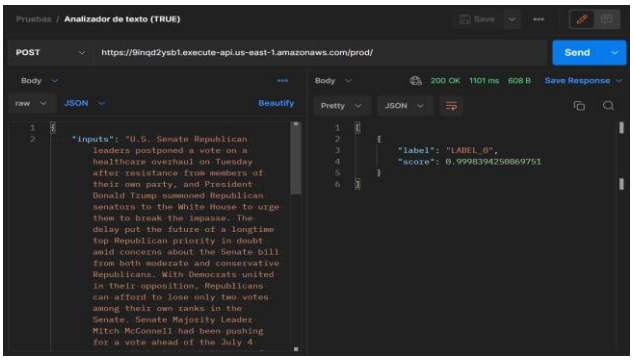

Ilustración 13: Evaluación del modelo para un valor "Real" del corpus

- Para una noticia considerada “Fake” en nuestro dataset:

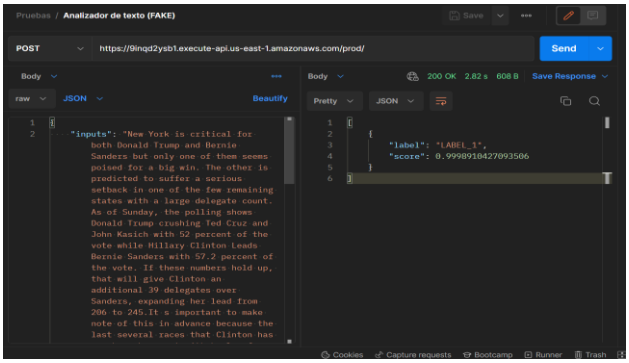

Ilustración 14: Evaluación del modelo para un valor considerado "Fake" en nuestro dataset

```
accuracy_score:
1.0
recall score:
1.0
confusion_matrix
[[213   0]
 [  0 207]]
```

Ilustración 15: Captura de las métricas obtenidas por nuestro modelo

Estos resultados en las métricas fueron posteriormente analizados ya que no se observaron con el corpus que se trabajó durante el desarrollo. Se llego a una hipótesis luego de analizar un conjunto de las noticias del corpus. Siendo esta que el modelo encontró una correlación entre el nombre del periódico y si la noticia es fake o no. Esto se podría verificar mediante el uso de Attention [12] pero escapa del alcance del proyecto.

XI.

VALIDACION PROTOTIPO

Característica	Definición o descirpción	1	2	3	4	5
Understandability	¿Fácil de comprender?					5
Documentation	¿Documentación de usuario completa, apropiada y bien estructurada?				4	
Buildability	¿Fácil de construir en un sistema compatible? (Close-Open)					5
Installability	¿Fácil de instalar en un sistema compatible?					5
Learnability	¿Fácil de aprender a usar sus funciones?					5
Identity	¿La identidad del proyecto / software es clara y única?					5
Copyright Licencing	¿Es fácil ver quién posee el proyecto / software?					5
	Adopción de la licencia apropiada?				4	
Governance	¿Fácil de entender cómo se ejecuta el proyecto y cómo se gestiona el desarrollo del software?					5
Community	¿Evidencia de comunidad actual / futura?					5
Accessibility	¿Evidencia de capacidad de descarga actual / futura?					5
Testability	¿Fácil de probar la corrección del funciones caja negra?					5
Portability	¿Utilizable en múltiples plataformas?					5
Supportability	¿Evidencia de soporte para desarrolladores actuales / futuros?					5
Analysability	¿Fácil de entender a nivel fuente?				4	
Changeability	¿Fácil de modificar y aportar cambios a los desarrolladores?					5
Evolvability	¿Evidencia de desarrollo actual / futuro?					5
Interoperability	¿Interoperable con otro software requerido / relaciona					5
modelo de evaluación basado en el estandar ISO 9126 ISO 15504 + ESCALA DE						

Ilustración 16: Tabla de evaluación del prototipo

En el ámbito de la asignatura Proyecto Final dos grupos de pares realizaron la validación del prototipo usando un estándar ISO. De esta manera se determinó la calidad del producto desarrollado y el resultado de dicha evaluación se puede observar en la ilustración 16.

XII.

CONCLUSIONES

El objetivo de este proyecto era el diseño de una herramienta que permitiera entrenar modelos de análisis de contenido. El objetivo de esto es la automatización del proceso de análisis de contenido para hacer clasificación de textos. Esto se hiso con la implementación de la herramienta en el jupyter notebook, siendo este la conclusión de una investigación y testeo de prototipos de diferentes modelos de Machine Learning para clasificación.

Este proyecto será útil ya que reduce la necesidad de humanos altamente entrenados para tareas de análisis de contenidos. Nuestro prototipo es fácil de implementar en varios contextos dependiendo del corpus con el que se entrene, pudimos cambiar de corpus en varias ocasiones sin tener que editar más que los parámetros mostrados en la ilustración 10 ahorrando tiempo de desarrollo y solucionando el problema de la confiabilidad que expresan en [13] al generar estándares constantes a la hora de desarrollar dicha tarea ya que una vez que los modelos son entrenados, los pesos de la red neuronal no cambian a diferencia de variables humanas que no podemos medir y tienden a cambiar en el tiempo.

Por otra parte, encontramos algunas limitaciones. Hacer un proceso tan genérico. La primera limitación es referente a los hiperparametros de los modelos durante el entrenamiento, esto se podría solucionar por medio de

exponer los parámetros a los usuarios finales, pero complicaría su uso para usuarios menos especializados. Se sacrifica eficiencia a cambio de facilidad de uso. Así mismo, es necesario que se haga todo el preprocesado de datos en una etapa previa como se evidencio con los resultados en las métricas del prototipo. Por último, la dependencia de una plataforma intermedia (Hugging Face) podría ser problemático en un futuro debido a cambios imprevistos en las políticas de esta.

Existen futuros caminos por explorar. Se podrían entrenar modelos para en vez de buscar la existencia o no de una categoría, se busque el porcentaje de aparición de una categoría en una secuencia de texto. Se podría intentar optimizar hiperparametros o agregar la opción de hacer una optimización de estos.

XIII. REFERENCIAS

- *T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv.org, 07-Sep-2013. [Online]. Available: https://arxiv.org/abs/1301.3781.*
- A. Mustansir, K. Shahzad, and M. K. Malik, "Towards Automatic Business Process Redesign: An NLP based approach to extract redesign suggestions - automated software engineering," *SpringerLink*, 03-Jan-2022. [Online]. Available: https://link.springer.com/article/10.1007/s10515-021-00316-8.
- *Ping Huang, Huijuan Zhu, Lei Zheng, and Ying Wang. 2021. Text Sentiment Analysis based on BERT and Convolutional Neural Networks. In 2021 5th International Conference on Natural Language Processing and Information Retrieval (NLPIR) (NLPIR 2021). Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3508230.3508231*

[1] Kuncahyo Setyo Nugroho, Anantha Yullian Sukmadewa, and Novanto Yudistira. 2021. Large-Scale News Classification using BERT Language Model: Spark NLP Approach. In 6th International Conference on Sustainable Information Engineering and Technology 2021 (SIET '21). Association for Computing Machinery, New York, NY, USA, 240–246. https://doi.org/10.1145/3479645.3479658

- *K Jia Ding, Yongjun Hu, and Huiyou Chang. 2020. BERT-Based Mental Model, a Better Fake News Detector. In Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence (ICCAI '20). Association for Computing Machinery, New York, NY, USA, 396–400. https://doi.org/10.1145/3404555.3404607*

[2] N. Arabadzhieva - Kalcheva and I. Kovachev, "Comparison of BERT and XLNet accuracy with classical methods and algorithms in text classification," 2021 International Conference on Biomedical Innovations and Applications (BIA), 2022, pp. 74-76, doi: 10.1109/BIA52594.2022.9831281

[3] *"ArXiv", Choice Reviews Online, vol. 45, n.º 02, pp. 45–0602—45–0602, octubre de 2007. Accedido el 9 de octubre de 2022. [En línea]. Disponible: https://doi.org/10.5860/choice.45-0602*

[4] *A. Mustansir, K. Shahzad y M. K. Malik, "Towards automatic business process redesign: an NLP based approach to extract redesign suggestions", Automated Software Engineering, vol. 29, n.º 1, enero de 2022. Accedido el 9 de octubre de 2022. [En línea]. Disponible: https://doi.org/10.1007/s10515-021-00316-8*

- *Fatma Elsafoury, Stamos Katsigiannis, Steven R. Wilson, and Naeem Ramzan. 2021. Does BERT Pay Attention to Cyberbullying? In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21). Association for Computing Machinery, New York, NY, USA, 1900–1904. https://doi.org/10.1145/3404835.3463029*
- *L. Deping, W. Hongjuan, L. Mengyang and L. Pei, "News text classification based on Bidirectional Encoder Representation from Transformers," 2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), 2021, pp. 137-140, doi: 10.1109/CAIBDA53561.2021.00036.*
- *W. Jing and Y. Bailong, "News Text Classification and Recommendation Technology Based on Wide & Deep-Bert Model," 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE), 2021, pp. 209-216, doi: 10.1109/ICICSE52190.2021.9404101.*
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html. [Accessed: 20-Nov-2022].
- S. Lacy, B. Watson, D. Riffe, and J. Lovejoy, "Issues and best practices in content analysis - sage journals," Sep-2015. [Online]. Available: https://journals.sagepub.com/doi/10.1177/1077699015607338. [Accessed: 20-Nov-2022].

R

Anexo

Titulo	Autores	Abstract	Keywords	Referencia
Does BERT Pay Attention to Cyberbullying?	Fatma Elsafoury, Stamos Katsigiannis, Steven R. Wilson, Naeem Ramzan	Social media have brought threats like cyberbullying, which can lead to stress, anxiety, depression, and in some severe cases, suicide attempts. Detecting cyberbullying can help to warn/ block bullies and provide support to victims. However, very few studies have used self-attention-based language models like BERT for cyberbullying detection and they typically only report BERT's performance without examining in depth the reasons for its performance. In this work, we examine the use of BERT for cyberbullying detection on various datasets and attempt to explain its performance by analyzing its attention weights and gradient-based feature importance scores for textual and linguistic features. Our results show that attention weights do not correlate with feature importance scores and thus do not explain the model's performance. Additionally, they suggest that BERT relies on syntactical biases in the datasets to assign feature importance scores to class-related wordsrather than cyberbullying-related linguistic features.	Cyberbullying, Text classification, BERT, NLP	Fatma Elsafoury, Stamos Katsigiannis, Steven R. Wilson, and Naeem Ramzan. 2021. Does BERT Pay Attention to Cyberbullying? In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21). Association for Computing Machinery, New York, NY, USA, 1900–1904. https://doi.org/10.1145/3404835.3463029
Text Sentiment Analysis based on BERT and Convolutional Neural Networks	Ping Huang, Huijuan Zhu, Lei Zheng, Ying Wang	The rapid development of the network has accelerated the speed of information circulation. Analyzing the emotional tendency contained in the network text is very helpful to tap the needs of users. However, most of the existing sentiment classification models rely on manually labeled text features, resulting in insufficient mining of deep semantic features hidden in the text, and it is difficult to improve the classification performance significantly. This paper presents a text sentiment classification model combining BERT and convolutional neural networks (CNN). The model uses BERT to complete the word embedding of the text, and then uses CNN to learn the deep semantic information about the text, so as mine the emotional tendency towards the text. Through verification on the large movie review dataset, BERT-CNN model can achieve an accuracy of 86.67%, which is significantly better than traditional classification method of textCNN. The results show that the method has good performance in this field.	BERT, Word embedding, Sentiment analysis, Convolutional Neural Networks	Ping Huang, Huijuan Zhu, Lei Zheng, and Ying Wang. 2021. Text Sentiment Analysis based on BERT and Convolutional Neural Networks. In 2021 5th International Conference on Natural Language Processing and Information Retrieval (NLPIR) (NLPIR 2021). Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3508230.3508231
Chinese Text Classification Method Based on BERT Word Embedding	Ziniu Wang, Zhilin Huang, Jianling Gao	In this paper, we enhance the semantic representation of the word through the BERT pre-training language model, dynamically generates the semantic vector according to the context of the character, and then inputs the character vector embedded as a character-level word vector sequence into the CapsNet.We builted the BiGRU module in the capsule network for text feature extraction, and introduced attention mechanism to focus on key information.We use the corpus of baidu's Chinese question and answer data set and only take the types of questions as classified samples to conduct experiments.We used the separate BERT network and the CapsNet as a comparative experiment. Finally, the experimental results show that the model effect is better than using one of the models alone, and the effect is improved.	Text Classification; BERT; CapsNet; Word Embedding; BiGRU; Attention mechanism	Ziniu Wang, Zhilin Huang, and Jianling Gao. 2020. Chinese Text Classification Method Based on BERT Word Embedding. In Proceedings of the 2020 5th International Conference on Mathematics and Artificial Intelligence (ICMAI 2020). Association for Computing Machinery, New York, NY, USA, 66–71. https://doi.org/10.1145/3395260.3395273
Large-Scale News Classification using BERT Language Model: Spark NLP Approach	Kuncahyo Setyo Nugroho, Anantha Yullian Sukmadewa, Novanto Yudistira	The rise of big data analytics on top of NLP increasing the computational burden for text processing at scale. The problems faced in NLP are very high dimensional text, so it takes a high computation resource. The MapReduce allows parallelization of large computations and can improve the efficiency of text processing. This research aims to study the effect of big data processing on NLP tasks based on a deep learning approach. We classify a big text of news topics with fine-tuning BERT used pre-trained models. Five pre-trained models with a different number of parameters were used in this study. To measure the efficiency of this method, we compared the performance of the BERT with the pipelines from Spark NLP. The result shows that BERT without Spark NLP gives higher accuracy compared to BERT with Spark NLP. The accuracy average and training	Large-scale text classification, distributed NLP architectures, BERT language model, Spark NLP	Kuncahyo Setyo Nugroho, Anantha Yullian Sukmadewa, and Novanto Yudistira. 2021. Large-Scale News Classification using BERT Language Model: Spark NLP Approach. In 6th International Conference on Sustainable Information Engineering and Technology 2021 (SIET '21). Association for Computing Machinery, New York, NY, USA, 240–246. https://doi.org/10.1145/3479645.3479658

		time of all model's using BERT is 0.9187 and 35 minutes while using BERT with Spark NLP pipeline is 0.8444 and 9 minutes. The bigger model will take more computation resources and need a longer time to complete the tasks. However, the accuracy of BERT with Spark NLP only decreased by an average of 5.7%, while the training time was reduced significantly by 62.9% compared to BERT without Spark NLP.		
Sentiment analysis on COVID tweets using COVID-Twitter-BERT with auxiliary sentence approach	Hung Yeh Lin, Teng-Sheng Moh	Sentiment analysis is a fascinating area as a natural language understanding benchmark to evaluate customers' feedback and needs. Moreover, sentiment analysis can be applied to understand the people's reactions to public events such as the presidential elections and disease pandemics. Recent works in sentiment analysis on COVID-19 present a domain-targeted Bidirectional Encoder Representations from Transformer (BERT) language model, COVID-Twitter BERT (CT-BERT). However, there is little improvement in text classification using a BERT-based language model directly. Therefore, an auxiliary approach using BERT was proposed. This method converts single-sentence classification into pair-sentence classification, which solves the performance issue of BERT in text classification tasks. In this paper, we combine a pre-trained BERT model from COVID-related tweets and the auxiliary-sentence method to achieve better classification performance on COVID tweets sentiment analysis. We show that converting single-sentence classification into pair-sentence classification extends the dataset and obtains higher accuracies and F1 scores. However, we expect a domain-specific language model would perform better than a general language model. In our results, we show that the performance of CT-BERT does not necessarily outperform BERT specifically in understanding sentiments.	Sentiment Analysis, Text Classification, BERT, Natural Language Processing, COVID, COVID-19, Tweets	Hung Yeh Lin and Teng-Sheng Moh. 2021. Sentiment analysis on COVID tweets using COVID-Twitter-BERT with auxiliary sentence approach. In Proceedings of the 2021 ACM Southeast Conference (ACM SE '21). Association for Computing Machinery, New York, NY, USA, 234–238. https://doi.org/10.1145/3409334.3452074
Website Category Classification Using Fine-tuned BERT Language Model	Ferhat Demirkıran; Aykut Çayır; Uğur Ünal; Hasan Dağ	The contents on the Word Wide Web is expanding every second providing web users a rich content. However, this situation may cause web users harm rather than good due to its harmful or misleading information. The harmful contents can contain text, audio, video, or image that can be about violence, adult contents, or any other harmful information. Especially young people may readily be affected with these harmful information psychologically. To prevent youth from these harmful contents, various web filtering techniques, such as keyword filtering, Uniform Resource Locator (URL) based filtering, Intelligent analysis, and semantic analysis, are used. We propose an algorithm that can classify websites, which may contain adult contents, with 67.81% (BERT) accuracy among 32 unique categories. We also show that a BERT model gives higher accuracy than both the Sequential and Functional API models when used for text classification.	Web filtering, BERT, Text classification, Func-tional API, Sequential API.	F. Demirkıran, A. Çayır, U. Ünal and H. Dağ, "Website Category Classification Using Fine-tuned BERT Language Model," 2020 5th International Conference on Computer Science and Engineering (UBMK), 2020, pp. 333-336, doi: 10.1109/UBMK50275.2020.9219384.
News Text Classification and Recommendation Technology Based on Wide & Deep-Bert Model	Wu Jing; Yang Bailong	With the rapid development of Internet technology, news information on various social platforms is growing wildly, generating large amounts of data. Since short text information such as news headlines, short messages, and newsletters has a small number of words and limited content, it is often difficult to extract effective information. The sparse feature information causes difficulties in text classification. If the news system cannot efficiently and accurately realize news classification and users preference recommendation, it will inevitably affect the experience and frequency of platform users. This paper mainly studies the application of deep learning in the field of text classification and users' personal recommendation. It uses multiple English text data sets to learn text features. Based on the Wide&Deep model, combined with the improved BERT pretraining model, the Wide&Deep-BERT model is designed. In addition, the corresponding news text classification and recommendation technology process is proposed, and the Tensorflow deep learning framework is used to experimentally verify the technology, which proves the effectiveness and practicability of the design technology.	Wide&deep model, best model, text classification	W. Jing and Y. Bailong, "News Text Classification and Recommendation Technology Based on Wide & Deep-Bert Model," 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE), 2021, pp. 209-216, doi: 10.1109/ICICSE52190.2021.9404101.
Text classification system of academic papers	Jin Dai; Chong Chen	The text classification is an important research orientation in the fields of information retrieval and data mining, which has extensive applications	Component; text classification; the academic text; BERT; BiGru	J. Dai and C. Chen, "Text classification system of academic papers based on hybrid Bert-BiGRU model," 2020 12th

based on hybrid Bert-BiGRU model		in the practical work and scientific research and its research on the algorithm is always a hot topic. At present, the study on the long text like academic texts mainly focuses on abstract extraction. Whereas, due to the complicated content of the text format, there is very little classification research according to the text structure. In this study, the academic texts are classified based on Bidirectional Encoder Representations from Transformers and Bidirectional Gated Recurrent Unit (BERT-BiGRU) model		International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2020, pp. 40-44, doi: 10.1109/IHMSC49165.2020.10088.
News text classification based on Bidirectional Encoder Representation from Transformers	Lin Deping; Wang Hongjuan; Liu Mengyang; Li Pei	In order to accurately and efficiently obtain information useful to us, people are paying more and more attention to the problem of data redundancy caused by excessive data information. In recent years, domestic and foreign researchers have proposed various frameworks for different natural language processing tasks, and different frameworks have different advantages and disadvantages. One of the classic problems in the field of natural language processing is text classification. News text classification is an important task that is easy to attract everyone's attention in our daily lives. This experiment is based on the BERT model under the Transformer framework to classify the news text data set. The same news text data set is compared with the RNN's long and short-term memory network. The evaluation index uses the general accuracy and loss value of the model classification. Experimental results show that the classification accuracy of the BERT model is significantly higher than that of the long and short-term memory network.	Component; news text classification; LSTM; Bert	L. Deping, W. Hongjuan, L. Mengyang and L. Pei, "News text classification based on Bidirectional Encoder Representation from Transformers," 2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), 2021, pp. 137-140, doi: 10.1109/CAIBDA53561.2021.00036.
Optimal Feature Selection for Imbalanced Text Classification	Anshu Khurana; Om Prakash Verma	Textual data suffers from two main problems; large number of features and class imbalance. Many conventional approaches and their variants exist in literature to solve both these problems. The classical Synthetic Minority Over-sampling Technique (SMOTE) is the most explored technique for balancing the dataset. We introduced a new algorithm to balance the dataset, named Distributed SMOTE (D_SMOTE), which overcomes the problem of lack of density and reducing the formation of small disjuncts. Further, another problem handled is the large number of features or high-dimensionality. To solve high-dimensionality, a novel feature selection technique is introduced known as modified Biogeography-Based Optimization (M_BBO). The proposed model, M_BBO, performs modification in ranking of variables using feature weighting algorithm rather than randomly ranking. We have proposed two new expressions in D_SMOTE and one new expression in M_BBO. The extensive experimental results are computed out on four text classification datasets with four machine learning classifiers. The results are concluded using three performance measures: 1) Area Under Curve, 2) G-mean and 3) F1-score. Our empirical and statistical observation for four class-imbalanced datasets shows that the proposed D_SMOTE outperforms the other similar oversampling technique. We have also compared our proposed algorithm, M_BBO+D_SMOTE, with other models on seventeen imbalanced text classification datasets. Our model outperformed the other models in fourteen datasets. We have also compared our model with Bidirectional Encoder Representations from Transformers (BERT). To validate the experimental analysis, statistical Friedman test is employed.	Text Classification, Distributed SMOTE, Modified Biogeography-Based Optimization, Class imbalance and Feature Selection	A. Khurana and O. P. Verma, "Optimal Feature Selection for Imbalanced Text Classification," in IEEE Transactions on Artificial Intelligence, doi: 10.1109/TAI.2022.3144651.
The Use of NLP-Based Text Representation Techniques to Support Requirement Engineering Tasks: A Systematic Mapping Review	R. Sonbol; G. Rebdawi; N. Ghneim	Natural Language Processing (NLP) is widely used to support the automation of different Requirements Engineering (RE) tasks. Most of the proposed approaches start with various NLP steps that analyze requirements statements, extract their linguistic information, and convert them to easy-to-process representations, such as lists of features or embedding-based vector representations. These NLP-based representations are usually used at a later stage as inputs for machine learning techniques or rule-based methods. Thus, requirements representations play a major role in determining the accuracy of different approaches. In this paper, we conducted a survey in the form of a systematic literature mapping (classification) to find out (1) what are the representations used in RE tasks literature, (2)	Natural language processing;requirements engineering;requirements representation;syntax;semantic	

		what is the main focus of these works, (3) what are the main research directions in this domain, and (4) what are the gaps and potential future directions. After compiling an initial pool of 2,227 papers, and applying a set of inclusion/exclusion criteria, we obtained a final pool containing 104 relevant papers. Our survey shows that the research direction has changed from the use of lexical and syntactic features to the use of advanced embedding techniques, especially in the last two years. Using advanced embedding representations has proved its effectiveness in most RE tasks (such as requirement analysis, extracting requirements from reviews and forums, and semantic-level quality tasks). However, representations that are based on lexical and syntactic features are still more appropriate for other RE tasks (such as modeling and syntax-level quality tasks) since they provide the required information for the rules and regular expressions used when handling these tasks. In addition, we identify four gaps in the existing literature, why they matter, and how future research can begin to address them.		
Utilizing a Rapidly Exploring Random Tree for Hazardous Gas Exploration in a Large Unknown Area	Y. A. Prabowo; B. R. Trilaksono; E. M. I. Hidayat; B. Yuliarto	The use of robotics olfaction for gas source localization or mapping has become a concern given the issues of terrorism or industrial accidents that may cause damage to the environment. A typical scenario is to send a robot to a place where a dangerous gas leak has just occurred. The robot's task is to map gas concentrations in the region of interest as effectively as possible. This paper addresses how the robot performs gas exploration in a large and unknown environment. One of the issues that needs to be addressed is the fact that the computation time of the path planning, frontier detection, goal decision making and gas distribution mapping is slower if all cells in the occupancy grid map are involved in a large environment. Consequently, the Rapidly-exploring Random Tree (RRT) algorithm is chosen as the main algorithm. The RRT graph guides the robot's navigation, utilizes the vertices as goal candidates, gas mean and variance value, and searches for a new frontier. A new strategy is proposed to address the frontier exploration and gas exploitation trade-off. Finally, a Robot Operating System (ROS), Gazebo, and a 3D gas simulator are used to compare the proposed strategy performance with the others in a large outdoor environment.	Robot olfaction;robot exploration;rapidly-exploring random trees	
Arabic Text Processing Model: Verbs Roots and Conjugation Automation	M. T. B. Othman; M. A. Al-Hagery; Y. M. E. Hashemi	The Natural Language Processing (NLP) is a process to automate the text or speech of Natural Languages. This automation is mainly conducted for Western languages. The Arabic Language got less focus in this area. This paper presents a Model to recognize an Arabic sentence. A new morphological model based on regular expressions is developed to recognize the Arabic verbs. A hash table containing all Arabic three-letters' root of verbs is implemented. The total number of Arabic verbs that are derived from three-letters' root size is 23090. The number of roots is 6104. A set of rules forming the Arabic grammar is used to derive and analyze the syntax of Arabic sentences. About 87% of the verbs represented in our regular expressions' engine are detected. Moreover, the sentences are also recognized. In several Surat of the Quran, only 9% of the detected verbs are false-positive (a non-verb declared as a verb), and 4% are considered false-negative (a verb is considered as a noun). This rate is mainly because we are not using vowels even that the Quran (our case study) is using them. The reason behind our decision is to be able to handle all Arabic texts, which mostly are not using vowels.	Arabic text processing;regular expression;root extraction;verbs root classification;data mining	
A3ID: An Automatic and Interpretable Implicit Interference Detection Method for Smart Home via Knowledge Graph	D. Xiao; Q. Wang; M. Cai; Z. Zhu; W. Zhao	The smart home brings together devices, the cloud, data, and people to make home living more comfortable and safer. Trigger-action programming enables users to connect smart devices using if-this-then-that (IFTTT)-style rules. With the increasing number of devices in smart home systems, multiple running rules that act on actuators in contradictory ways may cause unexpected and unpredictable interference problems, which can put residents and their belongings at risk. Previous studies have considered explicit interference problems related	Interference detection;knowledge graph;natural language processing (NLP);smart home	

		to multiple rules targeting a single actuator, whereas implicit interference (interference across different actuators) detection is still challenging and not yet well studied owing to the effort-intensive and time-consuming annotation work of obtaining device information. The lack of knowledge about devices is a critical reason that affects the accuracy and efficiency in implicit interference detection. In this article, we propose A3ID, an automatic detection method for implicit interference based on knowledge graphs. Using natural language processing (NLP) techniques and a lexical database, A3ID can extract knowledge of devices from a knowledge graph, including functionality, effect, and scope. Then, it analyzes and detects interferences among the different devices semantically in three steps, without human intervention. Furthermore, it provides user-friendly explanations in a well-designed structure to specify possible reasons for the implicit interference problems. Our experiment on 11 859 IFTTT-style rules shows that A3ID outperforms state-of-the-art methods by more than 33% in the F1-score for the detection of implicit interference. Moreover, evaluations on an extended data set for devices from ConceptNet (a knowledge graph) and five smart home systems suggest that A3ID also has favorable performance with other devices not limited to the smart home domain.		
TBLC-rAttention: A Deep Neural Network Model for Recognizing the Emotional Tendency of Chinese Medical Comment	Q. Jin; X. Xue; W. Peng; W. Cai; Y. Zhang; L. Zhang	In the current paper, a hybrid depth neural network model, TBLC-rAttention, aiming at Chinese text emotion recognition, is proposed to identify the emotional tendency of the Chinese medical reviews. The model includes the following steps: acquiring and preprocessing the Chinese corpus; mapping the preprocessed text into the word vectors; using Bi-directional Long Short-Term Memory network (Bi-LSTM) with the attention mechanism to acquire the context semantic features of the text; using Convolutional Neural Network (CNN) to obtain local semantics features on the basis of the context semantic features; and inputting the final feature vectors into the classification layer to complete the task of emotion recognition and the classification of the Chinese medical reviews. In this experiment, the corpus data is the comments of 999 cold medicine on a large e-commerce platform. All corpus are divided into three types, including high praise, medium praise and bad review. Classical machine learning models (SVM, NB) and neural network models (CNN, LSTM, Bi-LSTM, BiLSTM-Attention and RCNN) are performed as the comparison benchmarks to assess the category performance of TBLC-rAttention model. All the results were obtained when the training accuracy and test accuracy were stable after 1000 cycles of repeated calculation. The results show that TBLC-rAttention can get better text feature than the reference models, and the text classification accuracy reaches to 99%. In conclusion, the TBLC-rAttention model can identify semantic feature information to the greatest extent. In addition, this study also completes the numerical quantification of the predicted results.	Attention mechanism;bi-directional long short-term memory network (Bi-LSTM);Chinese medical comment;Chinese text emotion recognition;convolutional neural network (CNN);deep learning;feature extraction;hybrid neural network;I bayes (NB);natural language processing (NLP);support vector machine (SVM)	
"It's Like the Value System in the Loop": Domain Experts' Values Expectations for NLP Automation	Showkat, Dilruba and Baumer, Eric P. S.	The rise of automated text processing systems has led to the development of tools designed for a wide variety of application domains. These technologies are often developed to support non-technical users such as domain experts and are often developed in isolation of the tools primary user. While such developments are exciting, less attention has been paid to domain experts' expectations about the values embedded in these automated systems. As a step toward addressing that gap, we examined values expectations of journalists and legal experts. Both these domains involve extensive text processing and place high importance on values in professional practice. We engaged participants from two non-profit organizations in two separate co-speculation design workshops centered around several speculative automated text processing systems. This study makes three interrelated contributions. First, we provide a detailed investigation of domain experts' values expectations around future NLP systems. Second, the speculative	Values in Automated NLP, Natural Language Processing (NLP) Automation, Design Fiction, Participatory Design Fiction Workshop	Dilruba Showkat and Eric P. S. Baumer. 2022. "It's Like the Value System in the Loop": Domain Experts' Values Expectations for NLP Automation. In Designing Interactive Systems Conference (DIS '22). Association for Computing Machinery, New York, NY, USA, 100–122. https://doi.org/10.1145/3532106.3533483

		design fiction concepts, which we specifically crafted for these investigative journalists and legal experts, illuminated a series of tensions around the technical implementation details of automation. Third, our findings highlight the utility of design fiction in eliciting not-to-design implications, not only about automated NLP but also about technology more broadly. Overall, our study findings provide groundwork for the inclusion of domain experts values whose expertise lies outside of the field of computing into the design of automated NLP systems.		
Impact of Robotic Process Automation in Supply Chain: A Model for Task Selection	Rhouati, Abdelkader and Ettifouri, El Hassane and Dahhane, Walid and Abou Haidar, Georges	Robotic process automation (RPA) is one of the most emerging technology areas of the last decade. As the name implies, RPA is an approach to automate repetitive tasks in business operations. Many solutions are avail-able on the market by multiple vendors. Through the implementation of those RPA solutions, companies can achieve higher performance levels and lead a differentiating competitive edge. One of the first fields which have benefited from RPA is Supply Chain. This paper presents a solution for the task selection problem issue of RPA applied to the Supply Chain. A case study is also presented to demonstrate the effectiveness of the designed solution.	Robotic Process Automation, RPA, Supply Chain, viability assessment, User Interaction (UI) Logs, Machine Learning, Case Study, Task Selection, Business Process documentations	Abdelkader Rhouati, El Hassane Ettifouri, Walid Dahhane, and Georges Abou Haidar. 2021. Impact of robotic process automation in supply chain: A model for task selection. In 2021 the 3rd International Conference on Robotics Systems and Automation Engineering (RSAE) (RSAE 2021). Association for Computing Machinery, New York, NY, USA, 17–20. https://doi.org/10.1145/3475851.3475865
Content Discovery Using Perceptual Automation	Iliev, Alexander	In this work, an innovative media content discovery and selection system is proposed based on human behavior. There were two parts to this project: first, in the perception phase, a speech signal is used for analysis with the intention to detect and extract emotional cues; and second, in the automation phase, textual information was used to determine the sentiment using natural language processing methodology. The speech cues are first captured from the signal and then classified in three emotional categories: upbeat, positive or happy; emotionless, flat or neutral; and downbeat, negative or sad. Then in order to match the collected emotions from speech to specific contextual information captured from a real source, using natural language processing techniques five books from The Game of Thrones were processed. The results were summarized and mapping between the two systems was drawn so that books can be chosen based on the degree of each emotion portrayed from speakers. This can be done in a non-intrusive fission without direct and explicit human-machine intervention.	Content discovery selection emotion recognition speech glottal processing NLP perceptual automation	Alexander Iliev. 2018. Content discovery using perceptual automation. In Proceedings of the 10th International Conference on Management of Digital EcoSystems (MEDES '18). Association for Computing Machinery, New York, NY, USA, 233–238. https://doi.org/10.1145/3281375.3281399
Web Screen Reading Automation Assistance Using Semantic Abstraction	Ashok, Vikas and Puzis, Yury and Borodin, Yevgen and Ramakrishnan, I.V.	A screen reader's sequential press-and-listen interface makes for an unsatisfactory and often times painful web-browsing experience for blind people. To help alleviate this situation, we introduce Web Screen Reading Automation Assistant (SRAA) for automating users' screen-reading actions (e.g., finding price of an item) on demand, thereby letting them focus on what they want to do rather than on how to get it done. The key idea is to elevate the interaction from operating on (syntactic) HTML elements, as is done now, to operating on web entities (which are semantically meaningful collections of related HTML elements, e.g., search results, menus, widgets, etc.). SRAA realizes this idea of semantic abstraction by constructing a Web Entity Model (WEM), which is a collection of web entities of the underlying webpage, using an extensive generic library of custom-designed descriptions of commonly occurring web entities across websites. The WEM brings blind users closer to how sighted People perceive and operate on web entities, and together with a natural-language user interface, SRAA relieves users from having to press numerous shortcuts to operate on low-level HTML elements – the principal source of tedium and frustration. This paper describes the design and implementation of SRAA. Evaluation with 18 blind subjects demonstrates its usability and effectiveness.	Assistant, screen-reader, accessibility, natural interfaces, blind users, blindness, automation, web browsing	Vikas Ashok, Yury Puzis, Yevgen Borodin, and I.V. Ramakrishnan. 2017. Web Screen Reading Automation Assistance Using Semantic Abstraction. In Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI '17). Association for Computing Machinery, New York, NY, USA, 407–418. https://doi.org/10.1145/3025171.3025229
NLP-Based Enhancement of Information Security in ITO: A Diffusion of Innovation Theory Perspective	Bhatti, Baber Majid and Mubarak, Sameera and Nagalingam, Sev	Information technology outsourcing (ITO) has grown significantly in recent decades and is now over a USD trillion-dollar industry. Service provider organisations are striving to improve the efficiencies of their service deliveries. Natural language processing (NLP) provides an opportunity to bring efficiencies through automation in understanding and processing information. Since	Information security risk (ISR), information technology outsourcing (ITO), information security risk management (ISRM), natural language processing (NLP)	Baber Majid Bhatti, Sameera Mubarak, and Sev Nagalingam. 2020. NLP-based enhancement of information security in ITO: a diffusion of innovation theory perspective. In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASE '20). Association for

		information security risk management (ISRM) in ITO is a growing concern of both, client and service provider organisations, they are adopting to improve ISRM in ITO using NLP. This paper explores those ISRM improvement scenarios. It also investigates the information security risks (ISRs) that result from the use of NLP in ITO and proposes strategies to manage those ISRs. To gain insights into the problem, a qualitative research approach is followed using the case study method. Six semi-structured interviews were conducted from participants in three organisations in the ICT industry, engaged in an ITO relationship. To the best of our knowledge, it is the first study to investigate the use of NLP for enhancing ISRM in ITO.		Computing Machinery, New York, NY, USA, 112–117. https://doi.org/10.1145/3417113.3423373

Tabla 1: Tabla de Referencias