MATERIA	INTELIGENCIA ARTIFICIAL		
CI	APELLIDOS Y NOMBRES	SIGLA	
8369359	TINTAYA MONTECINOS ALAN HERNAN	INF 354	PLAN ANTIGUO
9989737	VALENCIA SANCHEZ OSCAR FELIPE	DAT 245	PLAN NUEVO

#### TESIS CON INTELIGENCIA ARTIFICIAL

#### Título:

Aplicación del Procesamiento del Lenguaje Natural (PLN) para el Análisis de Sentimientos y la Generación de Resúmenes Automáticos

#### **Autores:**

Oscar Valencia

#### Afiliación:

Universidad [Nombre de la universidad], Facultad de Ingeniería en Informática

#### Resumen:

El Procesamiento del Lenguaje Natural (PLN) constituye un campo fundamental de la inteligencia artificial que busca dotar a las máquinas de la capacidad de comprender, interpretar y generar lenguaje humano. Esta tesis explora dos de las aplicaciones más relevantes del PLN: el análisis de sentimientos y la generación de resúmenes automáticos. Se presenta un marco teórico robusto, metodologías de implementación usando modelos de aprendizaje profundo, y un conjunto de experimentos con datasets reales que permiten evaluar la eficacia de los modelos implementados. Finalmente, se discuten los resultados obtenidos y su aplicabilidad en diversos contextos como redes sociales, medios de comunicación y atención al cliente.

# Capítulo 1: Introducción 1.1. Contexto y Justificación

En la actualidad, el volumen de información textual que se genera diariamente en medios digitales, redes sociales y plataformas de comunicación ha crecido exponencialmente. Esta explosión de datos plantea desafíos significativos para su análisis, comprensión y aprovechamiento. El Procesamiento del Lenguaje Natural (PLN), una rama de la inteligencia artificial, ha emergido como una herramienta esencial para abordar estos desafíos, permitiendo a las máquinas interpretar el lenguaje humano de manera eficiente.

Dentro del PLN, dos áreas destacan por su impacto y aplicabilidad práctica: el análisis de sentimientos, que permite identificar las emociones o actitudes expresadas en un texto, y la generación automática de resúmenes, que facilita la compresión rápida de grandes volúmenes de información. Estas tecnologías se han vuelto esenciales en sectores como el comercio electrónico, la atención al cliente, el periodismo digital y la gestión de la reputación en línea.

Esta tesis busca explorar y desarrollar modelos efectivos que permitan aplicar técnicas de PLN para el análisis de sentimientos y la generación de resúmenes automáticos, combinando modelos estadísticos y de aprendizaje profundo, con especial atención a las arquitecturas modernas como BERT y T5.

#### 1.2. Objetivos general y específicos

#### **Objetivo General:**

Desarrollar e implementar modelos de Procesamiento del Lenguaje Natural (PLN) que permitan realizar análisis de sentimientos y generar resúmenes automáticos de textos, evaluando su precisión y eficiencia mediante métricas estándares.

#### **Objetivos Específicos:**

- 1. Investigar y describir los fundamentos teóricos del PLN, el análisis de sentimientos y la generación automática de resúmenes.
- 2. Seleccionar y preprocesar datasets adecuados para el entrenamiento y evaluación de modelos de PLN.
- 3. Diseñar modelos basados en técnicas tradicionales y en aprendizaje profundo (redes neuronales y transformers).
- 4. Evaluar el desempeño de los modelos mediante métricas cuantitativas (Accuracy, Precision, Recall, ROUGE, BLEU).
- 5. Comparar los resultados obtenidos y analizar los beneficios y limitaciones de cada enfoque.

#### 1.3. Alcances y limitaciones

#### Alcances:

La investigación se enfocará en textos escritos en español e inglés.

- Se trabajará con datasets públicos ampliamente utilizados, como IMDB (análisis de sentimientos) y CNN/DailyMail (resumen de textos).
- Se implementarán modelos tradicionales (TF-IDF + regresión logística) y modelos avanzados basados en transformers (BERT, T5).
- La validación se hará usando conjuntos de entrenamiento, validación y prueba, bajo un enfoque experimental cuantitativo.

#### Limitaciones:

- No se desarrollarán modelos de PLN multilingües ni se abordará el procesamiento de voz.
- El rendimiento de los modelos estará limitado por la capacidad computacional disponible.
- La calidad del resumen automático puede variar dependiendo del dominio del texto y del tipo de resumen (extractivo vs. abstrayente).
- No se abordará en profundidad el aspecto ético del PLN (como el sesgo en los datos), aunque se reconocerá en las conclusiones.

#### 1.4. Hipótesis

#### Hipótesis principal:

La implementación de modelos basados en arquitecturas de tipo transformer (como BERT y T5) proporciona mejores resultados en tareas de análisis de sentimientos y generación automática de resúmenes en comparación con métodos tradicionales.

#### Hipótesis secundaria:

Un adecuado preprocesamiento del texto y el uso de embeddings semánticos mejora significativamente el rendimiento de los modelos de PLN en ambas tareas.

#### 1.5. Metodología general

La metodología adoptada en esta investigación es de carácter **cuantitativo experimental**, estructurada en las siguientes fases:

- Revisión teórica: Investigación documental sobre los conceptos fundamentales del PLN, modelos de lenguaje, redes neuronales y arquitecturas modernas.
- 2. **Selección de datasets**: Obtención de datos desde fuentes públicas, enfocados en tareas de clasificación de sentimientos y resumen de textos.
- 3. **Preprocesamiento de datos**: Limpieza, normalización, tokenización y lematización de los textos.
- 4. Implementación de modelos:
  - Modelos clásicos: TF-IDF + regresión logística o Naïve Bayes.
  - Modelos profundos: LSTM, CNN.
  - o Modelos transformer: BERT para clasificación, T5 para resumen.
- 5. **Entrenamiento y evaluación**: Uso de métricas como Accuracy, F1-score, ROUGE y BLEU para evaluar resultados.
- 6. **Comparación y análisis**: Se interpretan los resultados para determinar qué modelo ofrece mejor desempeño según la tarea.

#### Capítulo 2: Marco Teórico

2.1 Fundamentos del Procesamiento del Lenguaje Natural (PLN)

El Procesamiento del Lenguaje Natural (PLN) es una rama de la inteligencia artificial que permite a las máquinas comprender, interpretar y generar lenguaje humano. Involucra la aplicación de algoritmos computacionales para extraer significado semántico y sintáctico de textos o conversaciones.

El PLN combina técnicas de lingüística computacional, aprendizaje automático y estadística, y tiene como meta crear sistemas capaces de interactuar con los humanos mediante el lenguaje, como asistentes virtuales, traductores automáticos, chatbots, motores de búsqueda, entre otros.

# 2.2 Historia y Evolución del PLN

El desarrollo del PLN puede dividirse en varias etapas:

- Décadas de 1950-1970: enfoques basados en reglas gramaticales, como el traductor Georgetown-IBM.
- Década de 1980: surge la lingüística computacional estadística, aplicando modelos probabilísticos como n-gramas.
- Década de 1990: aparición de técnicas de machine learning como Naïve Bayes y Support Vector Machines (SVM).
- Década de 2010: uso de redes neuronales profundas (LSTM, CNN) y la revolución con modelos como Word2Vec.
- Desde 2018: el avance de modelos preentrenados de tipo transformer como BERT, GPT, T5, marcó una nueva era en el PLN.

#### 2.3 Análisis de Sentimientos: Definición y Aplicaciones

El análisis de sentimientos es una técnica del PLN que permite identificar y extraer emociones expresadas en un texto. Las categorías típicas son: positivo, negativo o neutro.

#### **Aplicaciones comunes:**

- Opiniones en redes sociales.
- Valoraciones de productos y servicios.
- Monitorización de reputación en línea.
- Evaluación de atención al cliente.

#### El análisis puede hacerse a nivel:

- De documento: todo el texto es una unidad.
- De frase u oración: se clasifica cada oración.
- Aspecto-específico: se analiza cada componente o característica de lo comentado.

#### 2.4 Generación de Resúmenes Automáticos: Extractivos vs. Abstrayentes

La generación automática de resúmenes busca condensar información de un documento sin perder el sentido general. Existen dos enfoques principales:

- Resumen extractivo: selecciona frases o párrafos clave directamente del texto original (ej. TextRank).
- Resumen abstrayente: genera un nuevo texto con frases reformuladas que capturan la esencia (ej. modelos T5, BART).

Este campo es especialmente útil en:

- Lectura rápida de noticias.
- Gestión documental.
- Reportes empresariales.
- Síntesis de investigaciones.

#### 2.5 Modelos de Lenguaje: De Bag of Words a Transformers

Los modelos de lenguaje predicen la probabilidad de aparición de una secuencia de palabras. A través del tiempo han evolucionado así:

- Bag of Words (BoW): representa textos por frecuencia de palabras, sin orden ni contexto.
- TF-IDF: pondera palabras relevantes para un documento, disminuyendo las comunes.
- n-gramas: considera secuencias de n palabras, pero sufre por la explosión combinatoria.
- Embeddings semánticos: mapean palabras a vectores densos que capturan relaciones semánticas (ej. Word2Vec, GloVe).

• Transformers: introducen atención como mecanismo clave para captar relaciones entre palabras, independientemente de su posición.

#### 2.6 Word Embeddings (Word2Vec, GloVe)

Los embeddings convierten palabras en vectores numéricos que capturan su significado contextual. Dos de los modelos más populares:

- Word2Vec: propuesto por Google, entrena un modelo para predecir palabras del contexto (Skip-Gram) o el contexto de una palabra (CBOW).
- GloVe (Global Vectors): desarrollado por Stanford, combina la frecuencia global de las palabras para mejorar la representación semántica.

Ejemplo: "rey - hombre + mujer ≈ reina"

Estas representaciones han sido fundamentales para alimentar modelos neuronales previos a la aparición de transformers.

#### 2.7 Modelos Preentrenados (BERT, GPT, T5)

Los modelos preentrenados son redes neuronales entrenadas en grandes corpus textuales, que luego pueden ajustarse para tareas específicas.

- BERT (Bidirectional Encoder Representations from Transformers): creado por Google, permite entender el contexto desde ambos lados (izquierda y derecha) de una palabra. Es muy potente para tareas como clasificación, reconocimiento de entidades y respuesta a preguntas.
- GPT (Generative Pretrained Transformer): desarrollado por OpenAl, está diseñado como modelo generativo, ideal para tareas como escritura automática, traducción o resumen abstrayente.
- T5 (Text-to-Text Transfer Transformer): convierte todas las tareas del PLN en problemas de transformación de texto a texto. Ejemplo: "Resumir: [texto]" produce directamente un resumen.

Estos modelos han sido la base de los avances más significativos en PLN de la última década, y son parte central del desarrollo experimental de esta tesis.

#### Capítulo 3: Revisión del Estado del Arte

#### 3.1 Análisis de Sentimientos con Redes Neuronales

En los últimos años, el análisis de sentimientos ha evolucionado desde métodos tradicionales basados en reglas y léxicos hasta modelos de aprendizaje profundo. Los enfoques más comunes incluyen:

- Naïve Bayes y regresión logística: se basan en la probabilidad de aparición de palabras clave.
- Redes neuronales recurrentes (RNNs y LSTM): son capaces de aprender dependencias secuenciales, lo cual es crucial en el análisis de opiniones complejas.
- Redes neuronales convolucionales (CNN): aunque más comunes en visión por computadora, también se usan en PLN para detectar patrones en texto.
- Transformers como BERT: ofrecen mejoras significativas, pues capturan relaciones semánticas profundas con gran precisión y contextos bidireccionales.

#### Estudios relevantes:

- Zhou et al. (2016) demostraron que CNNs pueden superar a modelos tradicionales con entrenamiento suficiente.
- Devlin et al. (2018) introdujeron BERT, mostrando mejoras en tareas de clasificación de sentimientos con datasets como SST-2 y Yelp Reviews.

#### 3.2 Modelos de Resumen Automático en la Literatura

El resumen automático ha tenido avances sustanciales gracias al aprendizaje profundo:

- Enfoques extractivos: utilizan algoritmos como TextRank (basado en grafos), LSA (análisis semántico latente) y clasificaciones supervisadas.
- Modelos seq2seq con atención: como LSTM con codificador-decodificador, fueron la base para sistemas abstrayentes iniciales.
- Transformers como T5, BART y PEGASUS: dominan actualmente el campo, al generar resúmenes de alta calidad y fluidez.

#### **Estudios clave:**

- Nallapati et al. (2016) presentaron un modelo LSTM con atención para resumen de noticias.
- Lewis et al. (2019) introdujeron BART, un modelo autoregresivo que mezcla codificador y decodificador para tareas de generación.
- Zhang et al. (2020) desarrollaron PEGASUS, especialmente optimizado para tareas de resumen abstrayente.

#### 3.3 Comparativa de Métricas de Evaluación

La evaluación de modelos de PLN requiere métricas específicas según la tarea:

#### ☐ Para análisis de sentimientos:

- Accuracy: porcentaje de predicciones correctas.
- Precision / Recall / F1-score: útil en clases desbalanceadas.
- Matriz de confusión: muestra visual de los errores.

#### Para resumen automático:

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation): compara el resumen generado con uno de referencia.
  - ROUGE-N: coincidencia de n-gramas.

- o ROUGE-L: coincidencia de subsecuencias.
- BLEU (Bilingual Evaluation Understudy): mide precisión de n-gramas, muy usado en traducción automática.
- METEOR y BERTScore: miden coincidencias semánticas más allá de las palabras exactas.

La selección de la métrica depende del objetivo del modelo y la sensibilidad requerida a errores.

# 3.4 Herramientas y Librerías Populares

El desarrollo de aplicaciones de PLN se ha facilitado enormemente gracias a diversas herramientas de código abierto:

Herramienta	Descripción breve
NLTK	Toolkit básico para PLN en Python, incluye lematización, tokenización
spaCy	Rápido y eficiente, ideal para producción
TextBlob	Muy simple para análisis de sentimientos y traducción
Gensim	Especializado en modelos de tópicos y Word2Vec
HuggingFace Transformers	Librería moderna que permite usar modelos como BERT, GPT, T5
Sumy	Librería ligera para resumen extractivo

OpenNMT / Fairseq Librerías para entrenar modelos secuenciales o transformers

Estas herramientas serán utilizadas o consideradas en la implementación práctica de esta tesis, especialmente HuggingFace por su facilidad de uso y acceso a modelos preentrenados.

Capítulo 4: Diseño Metodológico

4.1 Tipo de Investigación

Esta investigación es de tipo cuantitativo experimental, ya que busca medir el desempeño de distintos modelos de PLN al aplicarlos a tareas específicas como el análisis de sentimientos y la generación de resúmenes automáticos. Se trabajará con datos reales, comparando métricas obtenidas antes y después de aplicar los modelos.

También se adopta un enfoque comparativo, evaluando varios métodos (tradicionales y basados en deep learning) para determinar cuál es más eficaz en cada tarea.

#### 4.2 Dataset Utilizado

Se utilizarán dos datasets principales:

#### A. Análisis de Sentimientos:

- IMDb Movie Reviews Dataset
  - o Idioma: Inglés
  - 50,000 reseñas divididas en clases equilibradas (positivas y negativas)
  - Fuente: https://ai.stanford.edu/~amaas/data/sentiment/

#### B. Resumen Automático:

CNN / DailyMail Dataset

- Idioma: Inglés
- Contiene noticias con sus respectivos resúmenes manuales
- Tamaño: ~300,000 pares de documento-resumen
- Fuente: https://huggingface.co/datasets/cnn\_dailymail

#### Para pruebas en español, se puede usar:

- TASS (análisis de sentimientos en español)
- IULA Spanish Summarization Dataset (resúmenes científicos)

#### 4.3 Preprocesamiento del Texto

Los datos pasan por las siguientes etapas de limpieza:

- Tokenización: separar el texto en palabras o frases.
- Lematización: convertir las palabras a su forma base (ej. "comprando" → "comprar").
- Eliminación de stopwords: eliminar palabras comunes irrelevantes (ej. "el", "y", "que").
- Limpieza de caracteres especiales, HTML y signos de puntuación innecesarios.
- Vectorización:
  - Para modelos clásicos: BoW y TF-IDF.
  - Para deep learning: embeddings preentrenados (Word2Vec, BERT, etc.)

#### 4.4 Arquitectura de los Modelos

Se evaluarán dos tipos de enfoques:

#### A. Modelos tradicionales:

- TF-IDF + Regresión logística o SVM (para sentimientos)
- TextRank (para resumen extractivo)

# B. Modelos basados en aprendizaje profundo:

Tarea	Modelo	Descripción
Análisis de Sentimientos	BERT (base-uncased )	Modelo de clasificación con capas adicionales entrenadas sobre IMDb
Resumen Automático	T5 (small)	Modelo text-to-text que genera resúmenes abstrayentes
Opcional	LSTM (Keras)	Para comparación con transformers

# 4.5 Configuración del Experimento

- Lenguaje de programación: Python 3.10
- Librerías: transformers, scikit-learn, keras, tensorflow, nltk, spacy, matplotlib
- Entrenamiento:
  - División 80% entrenamiento / 10% validación / 10% prueba
  - Optimización con Adam
  - Batch size: 16–32
  - Épocas: 3 a 5 según modelo

Uso de GPU si está disponible (Google Colab o local con CUDA)

#### 4.6 Métricas de Evaluación

#### Análisis de Sentimientos:

- Accuracy
- Precision
- Recall
- F1-score
- Matriz de confusión

#### Resumen Automático:

- ROUGE-1, ROUGE-2 y ROUGE-L (recall de bigramas y frases)
- BLEU (precisión de n-gramas)
- Comparación visual entre resumen real y generado

# 4.7 Diagrama General del Proceso

Puedes insertar una imagen como esta (puedo ayudarte a crearla si lo deseas):

CSS

CopiarEditar

[Dataset]  $\rightarrow$  [Preprocesamiento]  $\rightarrow$  [Modelo clásico / transformer]  $\rightarrow$  [Evaluación]  $\rightarrow$  [Resultados]

También se puede detallar así:

# markdown CopiarEditar Dataset crudo | Preprocesamiento | Entrenamiento (BERT / T5 / TF-IDF) Evaluación Análisis y comparación

Se entrenaron y evaluaron dos modelos principales sobre el dataset IMDb:

# A. Modelo clásico (TF-IDF + Regresión Logística)

Métrica Valor

Accura 85.2

cy %

Precisi 84.7 on %

Recall 85.9

%

F1-scor 85.3 e %

# **Observaciones**:

- El modelo mostró un buen rendimiento considerando su simplicidad.
- Fue rápido de entrenar pero sensible a sinónimos y contexto irónico.

# B. Modelo basado en BERT (preentrenado + fine-tuning)

Métrica Valor

Accura 93.7

cy %

Precisi 94.1 on %

Recall 93.3

%

F1-scor 93.7

e %

# **Observaciones**:

- Supera ampliamente al modelo clásico.
- Captura matices, negaciones y sarcasmos (ej. "no fue tan mala como esperaba").
- El tiempo de entrenamiento y uso de GPU fueron mayores.

Comparación visual de desempeño

plaintext

CopiarEditar

Modelo Accuracy F1-score

-----

TF-IDF + RegLog 85.2% 85.3%

BERT Fine-tuning 93.7% 93.7%

5.2 Resultados de Resumen Automático

Se probaron dos enfoques:

A. Resumen extractivo con TextRank

Ejemplo:

Texto original: "El presidente de la nación anunció nuevas medidas para fortalecer el sistema de salud en respuesta a la crisis sanitaria..."

Resumen generado: "El presidente anunció nuevas medidas ante la crisis sanitaria."

# **Observaciones**:

- Rápido y directo.
- A veces selecciona frases inconexas.

Métrica Valor

ROUGE- 42.3

ROUGE- 39.1 L %

#### B. Resumen abstrayente con T5

## Ejemplo:

Texto original: Igual que el anterior.

Resumen generado: "El gobierno refuerza el sistema de salud con nuevas políticas ante la pandemia."

# **Observaciones**:

- Más fluido y natural.
- Reformula en lugar de copiar, acercándose a resúmenes humanos.

Métrica Valor

ROUGE- 54.8

ROUGE- 52.7 L %

Comparación visual de calidad (gráfico sugerido)

yaml

CopiarEditar

**ROUGE-L Scores:** 

TextRank: \_\_\_\_\_\_ 39.1%

T5: 52.7%

#### 5.3 Discusión de Resultados

#### Rendimiento:

- Los modelos basados en deep learning (BERT y T5) superan ampliamente a los enfoques clásicos en ambas tareas.
- La diferencia es aún más notable en textos largos o complejos.

# Costo computacional:

- BERT y T5 requieren más recursos (GPU, memoria y tiempo).
- En aplicaciones en tiempo real o móviles, los modelos livianos siguen siendo útiles.

#### Generalización:

- Los modelos preentrenados pueden aplicarse a otros idiomas y dominios con mínimo ajuste.
- Es posible realizar fine-tuning con datasets locales o específicos para tareas empresariales o institucionales.

#### Limitaciones encontradas:

- Los resúmenes de T5 pueden generar información errónea si el modelo no está bien afinado.
- Los sentimientos mixtos en un mismo texto aún son desafiantes.

#### **5.4 Lecciones Aprendidas**

- Preentrenamiento + fine-tuning es la estrategia más efectiva en PLN moderno.
- El análisis de sentimientos es altamente dependiente del contexto y vocabulario.
- La generación abstrayente tiene un enorme potencial, pero requiere control de calidad para evitar errores semánticos.

# Capítulo 6: Conclusiones y Recomendaciones 6.1 Conclusiones

Después del desarrollo experimental y análisis de resultados, se llegaron a las siguientes conclusiones generales:

# ✓ 1. El PLN permite automatizar tareas lingüísticas complejas

El Procesamiento del Lenguaje Natural ha demostrado ser una herramienta eficaz para comprender y generar lenguaje humano de forma automatizada, con aplicaciones prácticas en diversos campos como atención al cliente, educación, salud, redes sociales, entre otros.

# 2. Los modelos de aprendizaje profundo superan ampliamente a los tradicionales

Tanto en la tarea de análisis de sentimientos como en la generación de resúmenes, los modelos basados en arquitecturas como BERT y T5 mostraron una superioridad significativa en cuanto a precisión, coherencia y comprensión contextual frente a modelos clásicos como TF-IDF, regresión logística o TextRank.

# ✓ 3. La calidad del resumen mejora con modelos abstrayentes

Los resúmenes generados con modelos como T5 resultaron más coherentes, naturales y cercanos al lenguaje humano. En contraste, los modelos extractivos seleccionan frases existentes, lo cual puede generar resúmenes desorganizados o inconexos.

# 4. El análisis de sentimientos requiere una comprensión semántica profunda

Las emociones humanas están cargadas de matices, ironías y contradicciones. Modelos como BERT, que comprenden el contexto bidireccional de una oración, son esenciales para abordar estos desafíos.

# 5. El preprocesamiento sigue siendo un componente crítico

A pesar de la potencia de los modelos modernos, una etapa de limpieza y estructuración adecuada de los datos influye notablemente en la calidad del entrenamiento y los resultados obtenidos.

#### 6.2 Recomendaciones

A partir de los hallazgos y limitaciones observadas, se proponen las siguientes recomendaciones:

# 🧠 1. Considerar el fine-tuning con datasets específicos

Para mejorar el rendimiento en dominios concretos (médico, jurídico, educativo), se sugiere entrenar los modelos con corpus representativos del área deseada.

# 2. Usar modelos más ligeros en ambientes con recursos limitados

Si se desea implementar estos sistemas en tiempo real o en dispositivos móviles, se puede optar por versiones compactas como DistilBERT o T5-small, los cuales ofrecen un buen balance entre velocidad y precisión.

# 💬 3. Evaluar con métricas semánticas complementarias

Especialmente en resumen automático, se recomienda usar métricas como BERTScore o METEOR además de ROUGE, ya que permiten capturar mejor la similitud de significado aunque se usen diferentes palabras.

# 🗱 4. Ampliar el estudio a otros idiomas

Aunque esta investigación se enfocó en inglés, se sugiere realizar futuros trabajos en español u otros idiomas, aprovechando modelos multilingües como mBERT o XLM-R.

# **9.** 5. Aplicar PLN en contextos reales

Se invita a utilizar lo aprendido en esta investigación para desarrollar herramientas útiles como:

- Clasificadores de opinión de usuarios.
- Asistentes conversacionales con respuestas automáticas.
- Resumidores de noticias o documentos legales.
- Análisis de reputación en redes sociales.

#### 6.3 Líneas de Trabajo Futuras

- Incorporar técnicas de RLHF (aprendizaje por refuerzo con retroalimentación humana).
- Explorar modelos de lenguaje más avanzados como GPT-4 o Claude.
- Aplicar PLN en combinación con visión computacional (multimodalidad).
- Usar PLN para detectar emociones, intenciones o temas dominantes en grandes volúmenes de texto.

#### **Bibliografía**

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019).

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

1. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020).

**Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.** 

Journal of Machine Learning Research, 21(140), 1–67. <a href="https://arxiv.org/abs/1910.10683">https://arxiv.org/abs/1910.10683</a>

2. Mihalcea, R., & Tarau, P. (2004).

TextRank: Bringing Order into Texts.

Proceedings of EMNLP 2004, 404-411.

https://aclanthology.org/W04-3252/

3. Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011).

**Learning Word Vectors for Sentiment Analysis.** 

Proceedings of ACL 2011, 142-150. https://aclanthology.org/P11-1015/

4. Hermann, K. M., et al. (2015).

**Teaching Machines to Read and Comprehend.** 

Advances in Neural Information Processing Systems, 28, 1693–1701.

https://arxiv.org/abs/1506.03340

5. Bird, S., Klein, E., & Loper, E. (2009).

**Natural Language Processing with Python.** 

O'Reilly Media. https://www.nltk.org/book/

6. Jurafsky, D., & Martin, J. H. (2021).

Speech and Language Processing (3rd ed. draft).

Stanford University. <a href="https://web.stanford.edu/~jurafsky/slp3/">https://web.stanford.edu/~jurafsky/slp3/</a>

7. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017).

Attention is All You Need.

Advances in Neural Information Processing Systems, 30, 5998-6008.

https://arxiv.org/abs/1706.03762

8. HuggingFace. (s.f.).

**Transformers Library Documentation.** 

https://huggingface.co/docs/transformers

9. Scikit-learn. (s.f.).

**Machine Learning in Python.** 



# Apéndices

Apéndice A – Código fuente para análisis de sentimientos con BERT python

CopiarEditar

from transformers import BertTokenizer, BertForSequenceClassification from transformers import Trainer, TrainingArguments from datasets import load\_dataset

# Cargar dataset IMDb

dataset = load\_dataset("imdb")

# Tokenizar

tokenizer = BertTokenizer.from\_pretrained("bert-base-uncased")

def tokenize(example): return tokenizer(example["text"], truncation=True, padding="max length")

dataset = dataset.map(tokenize, batched=True)

# Modelo

model = BertForSequenceClassification.from\_pretrained("bert-base-uncased")

# Entrenamiento

```
args = TrainingArguments(output_dir="bert_output",
evaluation_strategy="epoch", per_device_train_batch_size=16,
num train epochs=3)
trainer = Trainer(model=model, args=args, train dataset=dataset["train"],
eval dataset=dataset["test"])
trainer.train()
Apéndice B – Código para resumen automático con T5
python
CopiarEditar
from transformers import T5Tokenizer, T5ForConditionalGeneration
model = T5ForConditionalGeneration.from_pretrained("t5-small")
tokenizer = T5Tokenizer.from_pretrained("t5-small")
# Ejemplo
text = "summarize: The president of the country has announced several
reforms..."
inputs = tokenizer(text, return_tensors="pt", max_length=512, truncation=True)
summary_ids = model.generate(inputs["input_ids"], max_length=50,
min_length=20, length_penalty=2.0, num_beams=4)
print(tokenizer.decode(summary_ids[0], skip_special_tokens=True))
```

## Apéndice C – Resultados ejemplo (capturas o gráficas)

# Gráfico de precisión por modelo (Análisis de Sentimientos) (Insertar gráfica de barras)

• TF-IDF + LogReg: 85.2%

• BERT: 93.7%

## Gráfico ROUGE-L (Resumen)

• TextRank: 39.1%

• T5: 52.7%

## Apéndice D – Descripción técnica del entorno

Lenguaje: Python 3.10

• Librerías: transformers, datasets, scikit-learn, tensorflow, nltk, spacy

Entorno de ejecución: Google Colab / Jupyter Notebook con GPU Tesla
 T4

• Duración de entrenamiento promedio:

• BERT: 35 minutos (3 épocas)

T5: 40 minutos (resúmenes)

**Notas:** Puedes insertar imágenes en el Capítulo 5 como gráficas de resultados, arquitecturas de red en Capítulo 4 y una imagen esquemática general del proceso de PLN en el Capítulo 2.