



Universidad Michoacana de San Nicolás de Hidalgo

División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica

**OPTIMIZACIÓN DE MODELOS DE LENGUAJE  
MEDIANTE ALGORITMOS GENÉTICOS PARA LA  
DETECCIÓN DE SEXISMO EN TWEETS**

**TESIS**

Que para obtener el grado de

**MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA**

presenta

**Luis Miguel Huerta Venegas**

**Jaime Cerda Jacobo**

Director de Tesis

**José Ortiz Bejar**

Co-Director de Tesis

Morelia, Michoacán, Noviembre 2025



Universidad Michoacana de San Nicolás de Hidalgo  
Facultad de Ingeniería Eléctrica  
División de Estudios de Posgrado



**OPTIMIZACIÓN DE MODELOS DE LENGUAJE MEDIANTE  
ALGORITMOS GENÉTICOS PARA LA DETECCIÓN DE SEXISMO EN  
TWEETS**

Los Miembros del Jurado de Examen de Grado aprueban la Tesis de Maestría en Ciencias en Ingeniería Eléctrica de Luis Miguel Huerta Venezas

Dr. Juan Carlos Silva Chávez  
*Presidente del Jurado*

Dr. Jaime Cerda Jacobo  
*Director de Tesis*

Dr. José Ortiz Béjar  
*Co-director*

Dr. Luis Eduardo Gamboa Guzmán  
*Vocal*

Dr. Juan Carlos Olivares Rojas  
*Revisor Externo*

Dr. J. Aurelio Medina Rios  
*Jefe de la División de Estudios de Posgrado  
de la Facultad de Ingeniería Eléctrica. UMSNH  
(Por reconocimiento de firmas)*

UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO  
Noviembre 2025

## Agradecimientos

Agradezco profundamente al Dr. Jaime Cerda Jacobo y al Dr. José Ortiz Bejar por su guía constante, por ampliar el alcance de este trabajo y por su compromiso con mi formación académica. A mis colegas del trabajo y amistades cercanas, gracias por los debates, la ayuda técnica, las revisiones y la motivación que mantuvo vivo este proyecto.

Finalmente, quiero agradecer a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) por el impulso que brinda a los avances científicos en México; sin su apoyo este trabajo no hubiera sido posible. En particular, reconozco el apoyo proporcionado a través del Proyecto CF-2023-I-1174 dentro del programa “Ciencia de Frontera 2023”.



A mis padres, hermanos, asesores y compañeros de vida.



# Lista de Publicaciones

“Genetic Algorithm–Based Optimization of RoBERTa for Sexism Detection”

Luis Miguel Huerta-Venegas, Rodrigo Israel Hernandez-Marariegos, Jaime Cerda-Jacobo, José Ortiz-Bejar

Universidad Michoacana de San Nicolas de Hidalgo, Morelia, Mexico

Published in: 2025 IEEE Central America and Panama Convention (CONCA-PAN XLIII)





# Resumen

Este trabajo de tesis presenta una estrategia de optimización para un modelo de lenguaje aplicado al análisis de tweets basado en el conjunto de datos de la Tarea 1 del CLEF 2025 - “*EXIST: Sexism Identification and Characterization in Tweets*”. La estrategia propuesta combina el poder de los modelos de lenguaje preentrenados, como *RoBERTa*, con algoritmos genéticos (AGs) para la optimización de hiperparámetros. La investigación aborda tres subtareas: (1) Detección binaria de sexismo en tweets, (2) Caracterización de la intención del emisor, y (3) Categorización del tipo de sexismo.

El enfoque metodológico incluye una fase de análisis exploratorio de datos (*EDA*) para comprender la estructura del conjunto de datos y la distribución de clases, seguido de la selección del modelo base y la definición de los hiperparámetros a optimizar. El diseño del algoritmo genético involucra la representación de individuos, la definición de una función de aptitud basada en el *F1-Score* ponderado para las tres tareas, y la implementación de operadores genéticos para selección, cruce y mutación.

La integración de los AGs en el proceso de ajuste fino del modelo de lenguaje logró resultados significativos, superando configuraciones tradicionales en todas las métricas evaluadas. La evaluación del rendimiento se llevó a cabo mediante métricas tradicionales como *Precision*, *Recall* y *F1-Score*, demostrando mejoras que van desde el 2.57% hasta el 15.09% sobre métodos de ajuste fino estándar. Los hallazgos de esta investigación fueron publicados y presentados en el 2025 IEEE Central America and Panama Convention (CONCAPAN XLIII), demostrando el potencial de los AGs para mejorar el rendimiento de los modelos de lenguaje en tareas complejas de análisis de contenido en redes sociales.

Palabras clave: Optimización, Modelos de Lenguaje, Algoritmos Genéticos, *RoBERTa*, Detección de Sexismo.



# Abstract

This thesis presents an optimization strategy for a language model applied to tweet analysis, based on Task 1 of *CLEF 2025* - “*EXIST: Sexism Identification and Characterization in Tweets.*” The proposed strategy combines the power of pre-trained language models, such as *RoBERTa*, with Genetic Algorithms (GAs) for hyperparameter optimization. The research addresses three subtasks: (1) Binary sexism detection in tweets, (2) Source intention characterization, and (3) Sexism categorization.

The methodological approach includes an exploratory data analysis (*EDA*) phase to understand the dataset structure and class distribution, followed by the selection of the base model and the definition of hyperparameters to optimize. The genetic algorithm design involves the representation of individuals, the definition of a fitness function based on weighted *F1-Score* for the three tasks, and the implementation of genetic operators for selection, crossover, and mutation.

The integration of GAs into the language model fine-tuning process achieved significant results, outperforming traditional configurations in all evaluated metrics. Performance evaluation was conducted using traditional metrics such as *Precision*, *Recall*, and *F1-Score*, demonstrating improvements ranging from 2.57% to 15.09% over standard fine-tuning methods. The findings of this research were published and presented at the 2025 IEEE Central America and Panama Convention (CONCAPAN XLIII), demonstrating the potential of GAs to enhance language model performance in complex social media content analysis tasks.

Keywords: Optimization, Language Models, Genetic Algorithms, *RoBERTa*, Sexism Detection.



# Contenido

Agradecimientos . . . . .	III
Dedicatoria . . . . .	V
Lista de Publicaciones . . . . .	VII
Resumen . . . . .	IX
Abstract . . . . .	XI
Contenido . . . . .	XIII
Lista de Figuras . . . . .	XV
Lista de Tablas . . . . .	XIX
Lista de Símbolos . . . . .	XXI
1. Introducción . . . . .	1
1.1. Planteamiento del Problema . . . . .	1
1.2. Antecedentes . . . . .	2
1.3. Objetivos de la Tesis . . . . .	4
1.3.1. Objetivo general . . . . .	4
1.3.2. Objetivos particulares . . . . .	4
1.4. Justificación . . . . .	5
1.5. Alcances y Limitaciones . . . . .	5
1.6. Descripción de Capítulos . . . . .	6
2. Marco Teórico y Estado del Arte . . . . .	9
2.1. Detección de Sexismo y Discriminación en Texto . . . . .	9
2.2. Modelos de Lenguaje en PLN . . . . .	10
2.3. Redes Neuronales Artificiales (RNAs) . . . . .	14
2.3.1. Fundamentos sobre RNAs . . . . .	14
2.3.2. La Neurona Artificial . . . . .	15
2.3.3. Estructura de una RNA . . . . .	16
2.3.4. Funciones de Activación . . . . .	17
2.3.5. Entrenamiento de RNAs . . . . .	19
2.4. Algoritmos Genéticos (AGs) . . . . .	22
2.5. Brecha Actual y Oportunidades . . . . .	26
2.6. Comentarios Finales . . . . .	27

---

3. Metodología	29
3.1. Descripción del Conjunto de Datos	29
3.2. Estructura del Conjunto de Datos	31
3.3. Diseño de la Estrategia de Optimización	32
3.4. Métricas de Evaluación	34
3.5. Implementación	36
3.5.1. Entorno de Desarrollo y Herramientas	36
3.5.2. Entrenamiento y evaluación	37
3.6. Comentarios Finales	39
4. Resultados y Análisis de la Optimización con Algoritmos Genéticos	41
4.1. Análisis de Resultados	41
4.2. Comparación con Ajuste Fino Estándar	49
4.3. Comparación con los resultados de la competencia EXIST 2025	53
4.4. Comentarios Finales	55
5. Conclusiones	57
5.1. Logros Principales	57
5.2. Relevancia de la Integración de AGs en Modelos de Lenguaje	58
5.3. Limitaciones Identificadas	59
5.4. Implicaciones Prácticas	60
5.5. Reflexiones Finales	60
5.6. Impacto Académico y Participación en la Comunidad Científica	61
6. Trabajos Futuros	63
A. Evaluaciones Completas del Algoritmo Genético	65
Referencias	69

# Lista de Figuras

- 2.1. Modelo matemático de una neurona artificial. El diagrama ilustra el funcionamiento de una neurona artificial donde las señales de entrada  $(x_1, x_2, \dots, x_n)$  se multiplican por sus pesos correspondientes  $(w_1, w_2, \dots, w_n)$ , se suman para obtener la suma ponderada  $S = \sum_{i=1}^N w_i x_i$ , y luego se aplica una función de activación  $f(S)$  para generar la salida de la neurona. Este proceso simula el comportamiento de una neurona biológica, donde las sinapsis (pesos) modulan la transmisión de señales, y la función de activación determina si la neurona se activa o no, generando una respuesta no lineal que permite a la red aprender patrones complejos. . . . . 17
- 2.2. Arquitectura de una red neuronal artificial (RNA) de tres capas con alimentación hacia adelante (*feed-forward*). El diagrama muestra el flujo de información desde la capa de entrada (que recibe los datos originales), a través de una o más capas ocultas (que procesan y transforman la información mediante representaciones intermedias), hasta la capa de salida (que produce la respuesta final). Cada conexión entre neuronas tiene un peso asociado que se ajusta durante el entrenamiento mediante el algoritmo de retropropagación. Las capas ocultas permiten que la red aprenda representaciones jerárquicas y abstractas del problema, donde cada capa sucesiva captura patrones de mayor complejidad, habilitando así la capacidad de la red para modelar relaciones no lineales complejas entre los datos de entrada y salida. . . . . 18

2.3. Proceso de entrenamiento mediante el algoritmo de retropropagación (*back-propagation*) en una red neuronal. El diagrama ilustra el ciclo completo de aprendizaje: (1) Propagación hacia adelante (*forward pass*), donde los datos de entrada fluyen a través de las capas de la red, cada neurona calcula su salida aplicando la función de activación a la suma ponderada de sus entradas; (2) Cálculo del error en la capa de salida comparando la salida predicha con la salida deseada utilizando una función de pérdida (como error cuadrático medio o entropía cruzada); (3) Propagación hacia atrás (*backward pass*), donde el error se propaga desde la capa de salida hacia las capas ocultas, calculando el gradiente de la función de pérdida respecto a cada peso mediante la regla de la cadena; (4) Actualización de pesos, donde cada peso se ajusta en dirección opuesta al gradiente usando un algoritmo de optimización (como descenso por gradiente o *Adam*), minimizando así el error y mejorando el rendimiento de la red. Este proceso se repite iterativamente hasta que la red converge hacia una configuración óptima de pesos. . . . . 20

2.4. Flujo general de un algoritmo genético aplicado a la optimización de hiperparámetros. El diagrama muestra el ciclo evolutivo completo: (1) Inicialización de la población con individuos aleatorios codificando diferentes configuraciones de hiperparámetros; (2) Evaluación de cada individuo mediante la función de aptitud, que mide el rendimiento del modelo con esa configuración de hiperparámetros; (3) Selección de padres basándose en la aptitud, utilizando métodos como torneo o ruleta para favorecer individuos con mejor rendimiento; (4) Aplicación de operadores genéticos: cruza (*crossover*) que combina información de dos padres para generar descendencia, y mutación que introduce variaciones aleatorias para mantener diversidad; (5) Formación de nueva generación combinando individuos seleccionados y descendencia generada; (6) Reemplazo de la población anterior con la nueva generación; (7) Verificación de condición de parada (número máximo de generaciones o convergencia); si no se cumple, el ciclo se repite desde la evaluación. Este proceso iterativo permite explorar eficientemente el espacio de hiperparámetros, convergiendo hacia configuraciones óptimas que maximizan el rendimiento del modelo. . . . . 24

4.1. Evaluación de la función de aptitud durante 114 ejecuciones. . . . . 43

4.2. Comparación del valor promedio y el mejor valor de aptitud por generación del algoritmo genético. El gráfico muestra cómo ambas métricas evolucionan a lo largo de las 8 generaciones, revelando la convergencia del algoritmo y la estabilidad de la población. La proximidad entre el valor promedio y el mejor valor en cada generación indica la consistencia del rendimiento, mientras que la diferencia entre ambos refleja la diversidad de la población. La Generación 8 alcanza el mejor rendimiento absoluto (2.2404), y la tendencia general muestra una reducción progresiva en la diferencia entre promedio y mejor valor después de la Generación 2, indicando una mejora en la estabilidad y convergencia del algoritmo genético. . . . . 44



- 4.3. Análisis de estabilidad del algoritmo genético mediante el coeficiente de variación (CV) para cada generación. El gráfico muestra cómo el CV evoluciona a lo largo de las 8 generaciones, donde valores más bajos indican mayor estabilidad y consistencia en los resultados. La Generación 3 presenta el menor CV (0.006), demostrando la mayor estabilidad, mientras que la Generación 2 muestra el mayor CV (0.127) debido a la presencia de outliers. La tendencia general revela una mejora sustancial en la estabilidad después de la Generación 2, con todas las generaciones posteriores manteniendo CV por debajo de 0.021, lo que sugiere que el algoritmo ha alcanzado un estado de madurez donde produce resultados reproducibles y confiables. Esta estabilidad mejorada es crucial para garantizar que los hiperparámetros optimizados sean consistentes en implementaciones prácticas. . . . . 46



# Lista de Tablas

2.1. Comparación entre Funciones de Pérdida <i>MSE</i> y Entropía Cruzada . . . . .	21
3.1. Espacio de búsqueda de hiperparámetros para la optimización de <i>RoBERTa</i> mediante Algoritmos Genéticos . . . . .	33
3.2. Resumen de Métricas por Subtarea . . . . .	36
4.1. Evolución del valor de aptitud del algoritmo genético a través de 8 generaciones, mostrando la distribución de valores de aptitud para cada una de las 114 configuraciones evaluadas. El gráfico ilustra cómo el algoritmo converge progresivamente hacia valores superiores, evidenciando que las generaciones posteriores (especialmente la Generación 8) alcanzan valores de aptitud más altos (hasta 2.2404), mientras que las generaciones iniciales presentan mayor dispersión y valores más bajos. La mejora gradual en los valores mínimos y máximos de aptitud a lo largo de las generaciones demuestra la capacidad del algoritmo genético para explorar eficientemente el espacio de hiperparámetros y converger hacia configuraciones óptimas. . . . .	42
4.2. Resumen del Proceso de Optimización Evolutiva . . . . .	48
4.3. Comparación de configuraciones entre Ajuste- estándar y AG . . . . .	49
4.4. Comparación de Rendimiento: AG vs. Ajuste Fino Estándar . . . . .	50
4.5. Resumen estadístico de las métricas obtenidas con 30 ejecuciones en la etapa de evaluación final (corpus de prueba). CV = Coeficiente de Variación (Desviación Estándar/Promedio $\times$ 100). . . . .	51
4.6. Configuraciones Prometedoras por Generación del AG . . . . .	52
4.7. Comparación de Rendimiento: Modelo Optimizado (Corpus de Desarrollo) vs. Estado del Arte EXIST 2025 (Corpus de Prueba) . . . . .	54
A.1. Evaluaciones Completas del Algoritmo Genético (Parte 1) . . . . .	65
A.2. Evaluaciones Completas del Algoritmo Genético (Parte 2) . . . . .	66
A.3. Evaluaciones Completas del Algoritmo Genético (Parte 3) . . . . .	67



# Lista de Símbolos

## SÍMBOLOS DE REDES NEURONALES ARTIFICIALES

---

$x_i$	Entrada $i$ -ésima de una RNA.
$w_i$	Peso de conexión de la entrada $i$ -ésima.
$S$	Suma de entradas ponderadas: $S = \sum_{i=1}^N w_i x_i$ .
$f(S)$	Función de activación.
$\mathcal{L}_{MSE}$	Pérdida cuadrática media: $\mathcal{L}_{MSE} = \frac{1}{n} \sum (y_i - \hat{y}_i)^2$ .
$\mathcal{L}_{CE}$	Pérdida de entropía cruzada: $\mathcal{L}_{CE} = - \sum y_i \log(\hat{y}_i)$ .

---

## SÍMBOLOS DE ALGORITMOS GENÉTICOS

---

$\mathcal{P}^{(t)}$	Población en la generación $t$ .
$f(x_i)$	Función de aptitud del individuo $i$ .
$\mathcal{L}(x_i)$	Función de pérdida del individuo $i$ .
$k$	Tamaño del torneo.
$T$	Subconjunto para torneo.
$x_{\text{win}}$	Individuo ganador: $x_{\text{win}} = \arg \max_{x \in T} f(x)$ .
$\mathcal{N}(0, \sigma^2)$	Distribución normal con media 0 y varianza $\sigma^2$ .

---

## SÍMBOLOS DE OPTIMIZACIÓN Y HIPERPARÁMETROS

---

$lr$	Tasa de aprendizaje.
$d$	Tasa de abandono (dropout).
$b$	Tamaño de lote (batch size).
$w_1, w_2, w_3$	Pesos de pérdida de subtareas.
$n_{\text{layers}}$	Número de capas entrenables.

---

## SÍMBOLOS DEL MODELO MULTITAREA

---

$F1_{1.1}$	F1 de la tarea 1.1 (detección binaria).
$F1_{1.2}$	F1 de la tarea 1.2 (clasificación de intención).
$F1_{1.3}$	F1 de la tarea 1.3 (categorización).
$F1_{\text{weighted}}$	F1 ponderado: $F1_{\text{weighted}} = \sum_{i=1}^N w_i \cdot F1_i$ .
$F1_{\text{micro}}$	F1 micro: $F1_{\text{micro}} = \frac{2 \cdot \sum TP}{2 \cdot \sum TP + \sum FP + \sum FN}$ .
$F1_{\text{macro}}$	F1 macro: $F1_{\text{macro}} = \frac{1}{N} \sum_{i=1}^N F1_i$ .

---

## SÍMBOLOS MATEMÁTICOS GENERALES

---

$\arg \max$	Argumento que maximiza una función.
$\sum$	Sumatoria.
$\exp$	Exponencial.
$\log$	Logaritmo natural.
$\text{softmax}$	Activación softmax.
$\text{Softmax}_k$	Activación softmax para $k$ clases.

---

---

**ACRÓNIMOS GENERALES**


---

AG	Algoritmo Genético
ADAM	<i>Adaptive Moment Estimation</i> (Estimación Adaptativa de Momentos)
AI	<i>Artificial Intelligence</i> (Inteligencia Artificial)
BoW	<i>Bag of Words</i> (Bolsa de palabras)
CE	<i>Cross Entropy</i> (Entropía Cruzada)
CLEF	<i>Conference and Labs of the Evaluation Forum</i> (Conferencia y Laboratorios del Foro de Evaluación)
CNN	<i>Convolutional Neural Network</i> (Red Neuronal Convolutacional)
CPU	<i>Central Processing Unit</i> (Unidad Central de Procesamiento)
CUDA	<i>Compute Unified Device Architecture</i> (Arquitectura Unificada de Dispositivos de Cómputo)
CV	Coefficiente de Variación
DEAP	<i>Distributed Evolutionary Algorithms in Python</i> (Algoritmos Evolutivos Distribuidos en Python)
EDA	<i>Exploratory Data Analysis</i> (Análisis Exploratorio de Datos)
EmoBERT	<i>Emotion-aware BERT</i> (BERT Consciente de Emociones)
EXIST	<i>Sexism Identification and Characterization in Tweets</i> (Identificación y Caracterización de Sexismo en Tweets)
F1	<i>F1-Score</i> (Puntuación F1)
FN	<i>False Negatives</i> (Falsos Negativos)
FP	<i>False Positives</i> (Falsos Positivos)
TP	Verdaderos Positivos ( <i>True Positives</i> )
FIE	Facultad de Ingeniería Eléctrica
GPU	<i>Graphics Processing Unit</i> (Unidad de Procesamiento Gráfico)
GLUE	<i>General Language Understanding Evaluation</i> (Evaluación General de Comprensión del Lenguaje)
GRU	<i>Gated Recurrent Unit</i> (Unidad Recurrente con Puerta)
LSTM	<i>Long Short-Term Memory</i> (Memoria a Corto y Largo Plazo)
ML	<i>Machine Learning</i> (Aprendizaje Automático)
MSE	<i>Mean Squared Error</i> (Error Cuadrático Medio)
NLP	<i>Natural Language Processing</i> (Procesamiento del Lenguaje Natural)
PLN	Procesamiento del Lenguaje Natural
PyTorch	<i>Framework</i> de Aprendizaje Automático
ReLU	<i>Rectified Linear Unit</i> (Unidad Lineal Rectificada)
RMSprop	<i>Root Mean Square Propagation</i> (Propagación de Raíz Cuadrada Media)
RNA	Red Neuronal Artificial
RNC	Redes Neuronal Convolutacional
RNR	Red Neuronal Recurrente
GloVe	<i>Global Vectors for Word Representation</i> (Vectores Globales para Representación de Palabras)
GPT	<i>Generative Pre-trained Transformer</i> (Transformador Generativo Preentrenado)
BERT	<i>Bidirectional Encoder Representations from Transformers</i> (Representaciones de Codificador Bidireccional de Transformadores)
RoBERTa	<i>Robustly Optimized BERT Pretraining Approach</i> (Enfoque de Preentrenamiento de BERT Optimizado Robusto)
SGD	<i>Stochastic Gradient Descent</i> (Descenso por Gradiente Estocástico)
SVM	<i>Support Vector Machine</i> (Máquina de Soporte Vectorial)
TF-IDF	<i>Term Frequency-Inverse Document Frequency</i> (Frecuencia de Términos-Frecuencia Inversa de Documentos)
Word2Vec	<i>Word to Vector</i> (Palabra a Vector)

---

# Capítulo 1

## Introducción

### 1.1. Planteamiento del Problema

A pesar de los avances en modelos de lenguaje, la detección de sexismo y discriminación en texto sigue enfrentando desafíos significativos. Estos incluyen limitaciones en la optimización tradicional, donde los métodos basados en gradientes priorizan la minimización de pérdidas locales (métodos como Adam [Kingma14] o SGD), pero no garantizan una exploración global del espacio de soluciones. Asimismo, se presentan desbalances en los datos, ya que conjuntos como *EXIST 2025* [Basile25] suelen tener distribuciones desiguales de categorías de sexismo, afectando la generalización en tareas de detección de contenido sensible.

La integración de algoritmos genéticos presenta una opción para abordar estos problemas mediante la optimización adaptativa de hiperparámetros, ajustando dinámicamente: tasas de aprendizaje, capas de atención o funciones de pérdida para tareas multitarea de detección de discriminación. Además, permitiría la evolución de arquitecturas, diseñando automáticamente estructuras neuronales especializadas en capturar patrones discriminatorios sutiles. También contribuiría a la robustez ante datos ruidosos, utilizando operadores de mutación y cruce para mejorar la generalización en escenarios con datos limitados o desbalanceados.

Sin embargo, no existen estudios exhaustivos que evalúen la viabilidad de los AGs

(Algoritmos Genéticos) en este dominio ni comparen su rendimiento con métodos convencionales en el conjunto de datos de *EXIST* 2025 [Basile25]. La integración de algoritmos genéticos podría mitigar estos problemas al optimizar hiperparámetros críticos (tasa de aprendizaje, capas de atención, etc.) de forma adaptativa para tareas de detección de contenido sensible, evolucionar arquitecturas de modelos específicas para capturar patrones discriminatorios y prejuicios de género, y mejorar la robustez frente a datos ruidosos o escasos mediante estrategias de mutación y cruza controladas.

## 1.2. Antecedentes

La detección de discriminación y prejuicios en texto es una tarea fundamental en el procesamiento del lenguaje natural (PLN) con aplicaciones en áreas como la moderación de contenido, el análisis de redes sociales y la seguridad digital. Desde los trabajos pioneros de Wiebe et al. [Wiebe05a], quienes sentaron las bases para el análisis subjetivo del lenguaje, hasta los avances recientes en modelos de lenguaje preentrenados, el campo ha evolucionado significativamente. Sin embargo, la detección precisa de discriminación sigue siendo un desafío debido a la complejidad del lenguaje humano, la ambigüedad contextual y la diversidad cultural en la expresión discriminatoria. El trabajo de Mohammad y Kiritchenko [Mohammad18b] destacó la importancia de los conjuntos de datos bien anotados y la necesidad de modelos que capturen matices discriminatorios más allá de las categorías básicas (sexismo explícito, microagresiones, etc.). Por su parte, Téllez et al. [Téllez18b] y Graff et al. [Graff20] exploraron enfoques basados en aprendizaje profundo y transferencia de conocimiento, demostrando que los modelos preentrenados como *BERT* [Devlin19] y *GPT* [Radford18] pueden adaptarse a tareas de detección de discriminación específicas. No obstante, estos enfoques suelen depender de métodos de optimización basados en gradientes, que pueden quedar atrapados en óptimos locales y no explorar eficientemente el espacio de soluciones.

En 2024, Smith et al. [Smith24] propusieron *EmoBERT*, un modelo basado en *Transformer* optimizado para la detección de discriminación, que combina *embeddings* (representaciones vectoriales) contextuales y mecanismos de atención avanzados. Sin embargo, incluso estos



enfoques modernos enfrentan limitaciones en la exploración eficiente del espacio de hiperparámetros y en la adaptación a conjuntos de datos desbalanceados. En este contexto, los algoritmos genéticos (AGs) emergen como una alternativa prometedora para la optimización de modelos de lenguaje. Inspirados en la selección natural, los AGs permiten explorar espacios de búsqueda complejos, evitando óptimos locales y adaptándose dinámicamente a funciones de costo no diferenciables. Su aplicación en la configuración de hiperparámetros (tasa de aprendizaje, el número de capas en una red neuronal, el tamaño del conjunto de entrenamiento, etc. ), la selección de arquitecturas neuronales o incluso la evolución de *embeddings* podría ofrecer ventajas significativas en tareas de detección de discriminación.

## 1.3. Objetivos de la Tesis

### 1.3.1. Objetivo general

Diseñar y evaluar una estrategia de optimización basada en algoritmos genéticos con el propósito de mejorar el rendimiento de un modelo de lenguaje en la detección y caracterización de sexismo en texto, utilizando como conjunto de evaluación de referencia (*benchmark*) el conjunto de datos correspondiente a la Tarea 1 de CLEF-2025 “EXIST: Sexism Identification and Characterization in Tweets” [Basile25]. La investigación logró exitosamente demostrar mejoras significativas en todas las subtareas del desafío, superando el ajuste fino estándar con incrementos que van desde el 2.57 % hasta el 15.09 % en diferentes métricas, y obteniendo resultados que superan el décimo mejor resultado reportado en el desafío internacional *EXIST* 2025, destacándose especialmente mejoras del 13.2 % en *F1-score* para la clasificación de intención del emisor.

### 1.3.2. Objetivos particulares

Los objetivos particulares de esta investigación comprenden los siguientes aspectos: (1) analizar las limitaciones de los modelos tradicionales de optimización basados en gradientes [Wiebe05a, Graff20] y examinar los avances recientes en detección de discriminación y prejuicios en texto; (2) definir un marco metodológico sistemático para integrar algoritmos genéticos en la optimización de hiperparámetros durante el proceso de entrenamiento y ajuste fino de un modelo de lenguaje multitarea, considerando la selección del modelo base, la codificación de individuos, la configuración de operadores genéticos y la definición de funciones de aptitud; (3) implementar y validar la estrategia propuesta mediante la comparación de métricas de evaluación (*Precision*, *Recall*, *F1-Score*, *Accuracy*) con métodos basados en ajuste fino estándar en las tres subtareas de *EXIST* 2025, logrando mejoras consistentes en todas las métricas; y (4) identificar y caracterizar las ventajas y desventajas computacionales de los algoritmos genéticos en escenarios de recursos limitados para tareas de clasificación de contenido sensible en redes sociales, evaluando aspectos como la eficiencia computacional, el tiempo de convergencia y la reproducibilidad de los resultados.

## 1.4. Justificación

La integración de los trabajos de Wiebe et al. [Wiebe05a], Mohammad y Kiritchenko [Mohammad18b], Téllez et al. [Téllez18b] y Graff et al. [Graff20] proporciona un contexto teórico y práctico sólido para esta investigación. Estos estudios permiten realizar un análisis riguroso de la optimización evolutiva en modelos de lenguaje, proponiendo un marco innovador para abordar uno de los desafíos más persistentes en el procesamiento del lenguaje natural (PLN): la detección y caracterización de contenido discriminatorio y sexista en redes sociales.

## 1.5. Alcances y Limitaciones

La integración de los trabajos de Wiebe et al. [Wiebe05a], Mohammad y Kiritchenko [Mohammad18b], Téllez et al. [Téllez18b] y Graff et al. [Graff20] proporciona un contexto teórico y práctico sólido para esta investigación. Estos estudios permiten realizar un análisis riguroso de la optimización evolutiva en modelos de lenguaje, proponiendo un marco innovador para abordar uno de los desafíos más persistentes en el procesamiento del lenguaje natural (PLN): la detección y caracterización de contenido discriminatorio y sexista en redes sociales.

### Alcances

- El estudio se centrará en el conjunto de datos EXIST 2025 del desafío CLEF-2025, que incluye tweets con anotaciones sobre detección de sexismo, intención del emisor y categorización de tipos de discriminación.
- Se utilizará el modelo *RoBERTa* preentrenado como base para la optimización mediante algoritmos genéticos.
- Los algoritmos genéticos se aplicarán en la optimización de hiperparámetros críticos (tasa de aprendizaje, tasa de abandono, tamaño de lote y pesos de pérdida) para las tres subtareas simultáneas.

- Se evaluará el rendimiento en las tres subtarefas del desafío: detección binaria de sexismo, clasificación de intención del emisor y categorización de tipos de sexismo.

### Limitaciones

- La carga computacional de los algoritmos genéticos requiere múltiples evaluaciones de modelos completos, resultando en un costo computacional significativamente mayor que los métodos tradicionales.
- Los resultados están limitados al idioma inglés y al contexto de redes sociales, pudiendo no generalizar bien a otros idiomas o dominios.
- La efectividad del algoritmo genético depende críticamente de la definición de la función de aptitud, que requiere un diseño cuidadoso y puede no ser óptima para todos los escenarios.
- El enfoque puede enfrentar desafíos de escalabilidad cuando se aplica a modelos más grandes o conjuntos de datos más extensos.
- La interpretabilidad del modelo optimizado podría ser menor en comparación con enfoques más tradicionales o explícitos.

## 1.6. Descripción de Capítulos

El Capítulo 1, “Introducción y Planteamiento del Problema”, establece el contexto de la investigación, presenta el problema, revisa los antecedentes relevantes, define los objetivos, justifica el trabajo y delimita los alcances y limitaciones del estudio.

El Capítulo 2, “Marco Teórico y Estado del Arte en Detección de Discriminación y Algoritmos Genéticos”, proporciona el fundamento teórico necesario para comprender la investigación. Revisa los trabajos fundamentales en detección de discriminación en texto [Wiebe05a, Mohammad18b, Téllez18b, Graff20], presenta la evolución de los modelos de lenguaje desde enfoques tradicionales hasta arquitecturas modernas basadas en *Transformer* [Vaswani17], analiza modelos preentrenados como *BERT* [Devlin19], *GPT* [Radford18]

y *RoBERTa* [Liu19], y explora los algoritmos genéticos [Holland75, Goldberg89] como herramienta de optimización. Finalmente, identifica la brecha actual y las oportunidades de investigación.

El Capítulo 3, “Metodología”, describe el enfoque metodológico completo de la investigación. Presenta el conjunto de datos *EXIST* 2025 [Basile25], detalla el diseño de la estrategia de optimización evolutiva que integra algoritmos genéticos con el modelo *RoBERTa* [Liu19], define las métricas de evaluación utilizadas, y explica la implementación técnica que asegura la reproducibilidad de los experimentos.

El Capítulo 4, “Resultados y Análisis de la Optimización con Algoritmos Genéticos”, presenta los hallazgos experimentales de la investigación. Analiza el proceso de optimización evolutiva, la convergencia del algoritmo genético y la configuración óptima encontrada. Compara los resultados con métodos tradicionales de ajuste fino estándar y con el estado del arte reportado en el desafío *EXIST* 2025 [Basile25], demostrando mejoras significativas en todas las métricas evaluadas.

El Capítulo 5, “Conclusiones”, sintetiza los logros principales de la investigación, presenta las contribuciones al campo, discute las limitaciones identificadas y explora las implicaciones prácticas de los resultados obtenidos.

El Capítulo 6, “Trabajos Futuros”, propone líneas de investigación futuras que pueden expandir y mejorar las capacidades de los algoritmos genéticos en la optimización de modelos de lenguaje para la detección de contenido sensible, considerando extensiones a otros idiomas, optimizaciones multimodales, mejoras en eficiencia computacional y arquitecturas neuronales evolutivas.

El documento concluye con una sección de referencias bibliográficas y un apéndice con detalles técnicos del algoritmo genético implementado.



## Capítulo 2

# Marco Teórico y Estado del Arte

Este capítulo presenta los fundamentos conceptuales y las investigaciones relevantes que sustentan el desarrollo de una estrategia de optimización de modelos de lenguaje mediante algoritmos genéticos. Se exploran las bases teóricas de la detección de errores de anotación que se transmiten desde los etiquetadores humanos y discriminación en texto, los aportes clave en el área, las arquitecturas modernas de procesamiento del lenguaje natural (PLN), los vectores de atención como mecanismo fundamental, las redes neuronales artificiales (RNAs), y los algoritmos evolutivos. Finalmente, se describe la motivación de este trabajo.

### 2.1. Detección de Sexismo y Discriminación en Texto

Los fundamentos de la detección de contenido discriminatorio en texto consisten en identificar y clasificar automáticamente expresiones injustas o discriminatorias presentes en lenguaje escrito. Estas expresiones pueden manifestarse en diversas formas, como discriminación de género, racial, étnica, religiosa o de orientación sexual, siguiendo patrones de uso del lenguaje en un contexto social identificados en estudios realizados. La detección automática de este tipo de contenido se ha vuelto crucial en el análisis de redes sociales, la moderación de contenido y aplicaciones de seguridad digital.

Los aportes de Wiebe et al. [Wiebe05b] en el análisis subjetivo del lenguaje representan una contribución fundamental al introducir el concepto de análisis de subjetividad en

el procesamiento del lenguaje natural (PLN), permitiendo diferenciar entre contenido objetivo y subjetivo y mejorar así la identificación de expresiones discriminatorias. Por su parte, Mohammad y Kiritchenko [Mohammad18a] introdujeron el conjunto de datos EmoInt, que contiene *tweets* etiquetados con niveles de intensidad emocional discriminatoria, y propusieron métricas de evaluación como correlación de Pearson y F1 ponderada para medir el desempeño de los clasificadores.

Téllez et al. [Téllez18a] presentaron el conjunto de datos EMOTEX, diseñado para la clasificación de contenido discriminatorio en español, proponiendo un sistema basado en máquinas de soporte vectorial (SVM) que utiliza representaciones vectoriales como TF-IDF para identificar patrones en tareas de clasificación. Además, estudios recientes han explorado enfoques multimodales que integran texto, audio y video para la detección de contenido injusto, empleando arquitecturas de redes neuronales convolucionales y recurrentes para fusionar la información de distintas modalidades [Graff20].

## 2.2. Modelos de Lenguaje en PLN

Los modelos de lenguaje han experimentado una evolución revolucionaria en las últimas décadas, transformando el campo del procesamiento del lenguaje natural (PLN). Los primeros enfoques se apoyaron en modelos estadísticos básicos, principalmente en el uso de n-gramas para estimar probabilidades de secuencias de palabras. Si bien estos métodos ofrecieron una base inicial, enfrentaban limitaciones críticas como la dispersión de datos en secuencias largas, la incapacidad de capturar contexto semántico profundo y los inconvenientes de la dimensionalidad, que volvía intratables los modelos para órdenes elevados de n-grama. Posteriormente, la introducción de técnicas de *embeddings* como *Word2Vec* [Mikolov13] y *GloVe* [Pennington14] representó un avance sustancial al permitir representaciones densas y continuas de las palabras, capaces de reflejar similitudes semánticas y sintácticas. Sin embargo, estos modelos también presentaban limitaciones notables por ejemplo: la polisemia, al asignar un mismo vector a una palabra independientemente del contexto; la naturaleza estática de las representaciones, que no se ajustaban dinámicamente a diferentes usos; y la falta de una comprensión contextual completa a nivel oracional. Estos desafíos marcaron



---

la necesidad de arquitecturas más sofisticadas, que serían abordadas posteriormente con el surgimiento de los modelos basados en transformer.

La introducción de la arquitectura *Transformer* en 2017 [Vaswani17] revolucionó el PLN al eliminar la dependencia de las redes neuronales recurrentes (RNRs) y las redes de convolución (RNCs). Sus componentes clave incluyen la estructura encoder-decoder, que posibilita el procesamiento de secuencias de entrada y la generación de secuencias de salida correspondientes. La autoatención multi-cabeza permite el procesamiento paralelo mediante el cálculo de múltiples representaciones de atención de manera simultánea, cada una enfocada en distintos aspectos de la información. A su vez, las redes *feed-forward* proporcionan una transformación no lineal entre las capas de atención, mientras que la normalización por capas contribuye a estabilizar el entrenamiento al normalizar las activaciones. Finalmente, las conexiones residuales facilitan el flujo de gradientes en redes profundas, lo que hace posible un entrenamiento efectivo incluso en arquitecturas con un gran número de capas.

Los modelos basados en *transformer* han adoptado la estrategia de pre-entrenamiento en grandes corpus de texto, seguida de ajuste fino para tareas específicas. Entre los modelos más destacados se encuentra *BERT* (*Bidirectional Encoder Representations from Transformers*) [Devlin19], que utiliza pre-entrenamiento bidireccional con tareas de enmascaramiento de lenguaje, captura contexto bidireccional completo, y es efectivo para tareas de clasificación y extracción de información. *GPT* (*Generative Pre-trained Transformer*) [Radford18] emplea pre-entrenamiento unidireccional con predicción de lenguaje, siendo excelente para generación de texto y completado de frases, aunque limitado en tareas que requieren contexto bidireccional. *RoBERTa* (Robustly Optimized *BERT* Pretraining Approach) [Liu19] optimiza el pre-entrenamiento de *BERT* con mayor tamaño del lote de entrenamiento y mejor optimización de hiperparámetros, logrando mejores resultados en múltiples benchmarks de PLN.

El mecanismo de atención se fundamenta en la idea de que no todas las palabras de una secuencia poseen la misma relevancia para comprender su significado global. Desde una perspectiva formal, la atención se define como una función que mapea una consulta (*query*) y un conjunto de pares clave-valor (*key-value*) a una salida. Dicha salida se obtiene como una suma ponderada de los valores, en la que los pesos asignados a cada uno se determinan

mediante una función de compatibilidad que evalúa la relación entre la consulta y las claves correspondientes.

La función de atención se expresa matemáticamente como:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.1)$$

donde  $Q \in \mathbb{R}^{n \times d_k}$  es la matriz de consultas,  $K \in \mathbb{R}^{m \times d_k}$  es la matriz de claves,  $V \in \mathbb{R}^{m \times d_v}$  es la matriz de valores,  $d_k$  es la dimensión de las claves, y  $n$  y  $m$  son las longitudes de las secuencias de entrada y salida.

El mecanismo de atención incorpora un factor de escalado, representado como  $\sqrt{d_k}$ , que resulta fundamental para estabilizar los gradientes durante el entrenamiento, evitando que los valores de la función softmax tiendan hacia cero o uno. Este mecanismo puede entenderse como un proceso compuesto por tres pasos principales: primero, la comparación, en la que la consulta se contrasta con cada clave para determinar su relevancia; segundo, la ponderación, donde los puntajes de compatibilidad obtenidos se normalizan mediante softmax; y tercero, la agregación, en la que los valores se combinan utilizando los pesos calculados. Gracias a esta formulación, el modelo puede “enfocarse” en diferentes partes de la entrada de acuerdo con el contexto, de manera análoga a la atención selectiva que los seres humanos aplican al procesar información.

Los modelos modernos emplean distintas variantes del mecanismo de atención para capturar relaciones lingüísticas y contextuales. La autoatención permite detectar dependencias de largo alcance y revelar prejuicios implícitos, como los de género. La atención multi-cabeza enriquece este proceso al enfocarse en aspectos sintácticos, semánticos, contextuales y temáticos, lo que facilita identificar tanto discriminaciones explícitas como microagresiones sutiles. Por último, la atención cruzada en arquitecturas *encoder-decoder* resulta útil para comparar textos o documentos, fortaleciendo la detección de patrones discriminatorios más complejos. La implementación de mecanismos de atención en arquitecturas *Transformer* requiere técnicas como proyecciones lineales, normalización, conexiones residuales y estrategias de entrenamiento especializadas (incremento gradual de la tasa de aprendizaje, programación adaptativa y recorte de gradientes) para garantizar estabilidad y eficiencia. Estas herramientas permiten que los vectores de atención se utilicen en la detección de se-

---

xismo y discriminación en texto, facilitando la identificación de palabras o frases relevantes, el análisis de contextos y patrones de co-ocurrencia, y aportando transparencia y confiabilidad a las decisiones del modelo en aplicaciones sensibles. Además, los modelos de atención permiten realizar un análisis detallado de casos específicos: el estudio de falsos positivos explica por qué ciertos textos se clasifican incorrectamente, mientras que el de falsos negativos revela situaciones donde el modelo no logra identificar prejuicios, evidenciando limitaciones en su capacidad de detección. Por último, el análisis de casos límite examina textos situados entre diferentes categorías, proporcionando información sobre la sensibilidad del modelo y las áreas que requieren mejoras adicionales.

A pesar de sus ventajas, los modelos de atención presentan desafíos significativos que deben considerarse. En términos computacionales, la atención cuadrática de los *Transformers* limita la escalabilidad, ya que el costo crece cuadráticamente con la longitud de la secuencia, lo que genera altos requerimientos de memoria, tiempos de entrenamiento prolongados y una inferencia más lenta, especialmente en textos largos o en dispositivos con recursos limitados. Además, los modelos pueden heredar y amplificar desigualdades reflejadas en los datos presentes en los datos de entrenamiento, la amplificación ocurre cuando la atención se centra en patrones discriminatorios presentes en los datos, y los errores de anotación se transmiten desde los etiquetadores humanos, lo que requiere procesos de anotación más cuidadosos. Por último, persisten desafíos de interpretabilidad: los patrones de atención pueden ser complejos y difíciles de comprender en modelos grandes, muestran inestabilidad entre ejecuciones, lo que obliga a realizar análisis adicionales para interpretar correctamente las decisiones del modelo

La comprensión y optimización de los vectores de atención es fundamental para mejorar el rendimiento y la interpretabilidad de los modelos. Parámetros como el número de cabezas, la dimensionalidad, la profundidad de capas y la tasa de abandono determinan la capacidad de capturar relaciones complejas, mientras que métricas de atención permiten evaluar consistencia, diversidad e interpretabilidad. Integrar estos parámetros en algoritmos genéticos facilita la exploración eficiente de configuraciones óptimas que maximicen tanto el desempeño como la claridad de las decisiones del modelo.

Los modelos de atención encuentran aplicaciones específicas en la identificación de

contenido discriminatorio gracias a su capacidad para comprender el contexto y desambiguar significados, incluso frente a ironía, sarcasmo o ambigüedad. Los modelos preentrenados pueden personalizarse mediante ajuste fino para tareas concretas, como la detección de sexismo en redes sociales. Son particularmente eficaces para identificar prejuicios de género al conectar palabras relacionadas con el género con características estereotípicas, detectando patrones implícitos y microagresiones que refuerzan roles tradicionales, mientras que el análisis contextual permite distinguir críticas legítimas de expresiones discriminatorias. De manera similar, estos modelos son valiosos para detectar prejuicios raciales y étnicos, identificando estereotipos, discriminación institucional y evaluando la representación equitativa de diferentes grupos, lo que proporciona métricas útiles para auditar la inclusividad del contenido y revelar patrones discriminatorios profundamente arraigados en el lenguaje.

## 2.3. Redes Neuronales Artificiales (RNAs)

### 2.3.1. Fundamentos sobre RNAs

Las redes neuronales artificiales (*Artificial Neural Networks*, RNAs) son modelos computacionales inspirados en la estructura y funcionamiento del sistema nervioso de los seres vivos. Han sido ampliamente utilizadas para modelar la relación entre datos de entrada y salida, especialmente en escenarios donde dichas relaciones presentan una naturaleza no lineal o son difíciles de formalizar mediante métodos tradicionales.

Se denominan “redes” porque interconectan múltiples unidades de procesamiento simples, denominadas neuronas artificiales, que trabajan de forma conjunta. Esta arquitectura distribuida permite que las RNAs exhiban un comportamiento colectivo complejo que simula, de manera abstracta, el procesamiento de la información en el cerebro humano o animal. Aunque el nivel de detalle biológico es altamente simplificado, esta abstracción resulta suficiente para capturar patrones en datos complejos y dinámicos. Una característica distintiva de las RNAs es su capacidad de aprendizaje. Es decir, pueden aprender a resolver tareas específicas a partir de ejemplos, y una vez entrenadas, tienen la habilidad de generalizar su conocimiento a nuevas instancias similares. Este principio de generalización es clave en aplicaciones prácticas, ya que permite que una red resuelva problemas no vistos durante

el entrenamiento. Para que una RNA pueda ser utilizada en la práctica, primero debe ser entrenada. Este entrenamiento se lleva a cabo mediante un algoritmo de aprendizaje supervisado, como el algoritmo de retropropagación del error, que actúa como un “profesor” para la red. En este proceso, se presentan datos de entrada junto con sus respectivas salidas deseadas, y el algoritmo ajusta los pesos de conexión entre las neuronas con base en el error cometido por la red. Estos pesos suelen ser inicializados con valores aleatorios pequeños, lo que implica que el comportamiento de la red es no determinista, ya que resultados distintos pueden emerger entre ejecuciones si no se fija una semilla aleatoria. Desde una perspectiva matemática, las RNAs han sido reconocidas como aproximadores universales, lo que significa que pueden aproximar cualquier función continua con un error arbitrariamente pequeño, siempre que cuenten con una arquitectura y parámetros adecuados. Gracias a esta propiedad, las redes neuronales son especialmente efectivas para modelar sistemas complejos donde existen múltiples interacciones no lineales entre variables. Las RNAs ofrecen un enfoque potente y flexible frente a métodos convencionales en tareas de predicción, clasificación, modelado de comportamiento y reconocimiento de patrones, posicionándolas como una herramienta clave en el campo del aprendizaje automático y la inteligencia artificial. Son la base sobre la cual se construyen arquitecturas más complejas como las redes convolucionales (RNCs) y las recurrentes (RNRs), predecesoras directas de los “*Transformer*”.

### 2.3.2. La Neurona Artificial

En el núcleo funcional de una red neuronal artificial se encuentra la neurona artificial, también conocida como unidad de procesamiento. Este componente simula, de manera abstracta, el comportamiento de una neurona biológica. Una red neuronal artificial (RNA) se construye interconectando múltiples de estas neuronas, formando una estructura capaz de aprender representaciones complejas a partir de los datos.

En la estructura biológica, las neuronas están conectadas entre sí mediante sinapsis, que transmiten señales electroquímicas. En una RNA, estas conexiones sinápticas son representadas por conexiones entre neuronas que transportan información en forma de señales numéricas (valores reales). Cada conexión posee un peso sináptico, que interactúa con la señal de entrada asociada, y cuya adaptación a lo largo del proceso de entrenamiento

permite a la red aprender, este comportamiento se muestra por la fórmula (2.1),

$$S = \sum_{i=1}^N w_i x_i \quad (2.2)$$

Donde  $x_i$  es el valor de la entrada  $i$  y  $w_i$  es el peso de la conexión.

Una neurona artificial típica recibe una o más señales de entrada (inputs), cada una multiplicada por su peso correspondiente. Luego, calcula una combinación lineal de esas entradas ponderadas y le aplica una función de activación (Ecuación 2.3) para generar una salida (output). Este output puede a su vez ser utilizado como entrada para otras neuronas en la red. En la Figura 2.1 se muestra el modelo de este comportamiento.

$$k = f(S) \quad (2.3)$$

Así, la neurona artificial actúa como una unidad básica de transformación de datos, permitiendo a la red aprender patrones complejos cuando se agrupan múltiples neuronas en capas y se estructuran jerárquicamente. En conjunto, estas unidades de procesamiento cooperan para emular la estructura interna y la capacidad de aprendizaje del sistema neuronal de los animales, aunque en una forma mucho más simplificada.

### 2.3.3. Estructura de una RNA

Una red neuronal artificial (RNA) está conformada por múltiples unidades de procesamiento denominadas neuronas artificiales. De forma análoga a las neuronas biológicas, una sola neurona artificial tiene capacidades limitadas para resolver tareas complejas; no obstante, el verdadero poder computacional de una RNA surge de la interacción y conexión entre un gran número de estas unidades.

Las neuronas se organizan estructuralmente en capas: una capa de entrada que recibe los datos originales, una o varias capas ocultas que procesan la información de manera jerárquica, y una capa de salida que produce la respuesta final. Aunque las neuronas de una misma capa comparten las mismas entradas, cada una puede generar salidas diferentes debido a que poseen pesos sinápticos únicos y, en algunos casos, funciones de activación distintas.

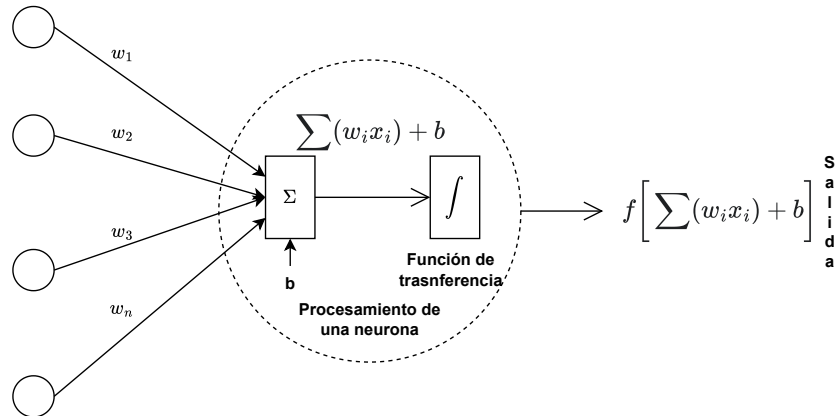


Figura 2.1: Modelo matemático de una neurona artificial. El diagrama ilustra el funcionamiento de una neurona artificial donde las señales de entrada  $(x_1, x_2, \dots, x_n)$  se multiplican por sus pesos correspondientes  $(w_1, w_2, \dots, w_n)$ , se suman para obtener la suma ponderada  $S = \sum_{i=1}^N w_i x_i$ , y luego se aplica una función de activación  $f(S)$  para generar la salida de la neurona. Este proceso simula el comportamiento de una neurona biológica, donde las sinapsis (pesos) modulan la transmisión de señales, y la función de activación determina si la neurona se activa o no, generando una respuesta no lineal que permite a la red aprender patrones complejos.

Uno de los modelos más representativos es la red neuronal de alimentación hacia adelante (*feed-forward*) de tres capas, en la cual la información fluye unidireccionalmente desde la entrada hacia la salida sin ciclos ni retroalimentación. Las capas ocultas desempeñan un papel crucial al permitir que la red aprenda representaciones intermedias y abstractas del problema. A mayor profundidad (es decir, más capas ocultas), mayor es la capacidad de la red para modelar patrones y relaciones no lineales complejas. En la Figura 2.2 se ilustra este modelo clásico de red neuronal.

### 2.3.4. Funciones de Activación

Las funciones de activación son componentes fundamentales en redes neuronales, ya que introducen no linealidad y permiten aprender patrones complejos y límites de decisión no lineales. Cada neurona aplica una función de activación a la suma ponderada de sus

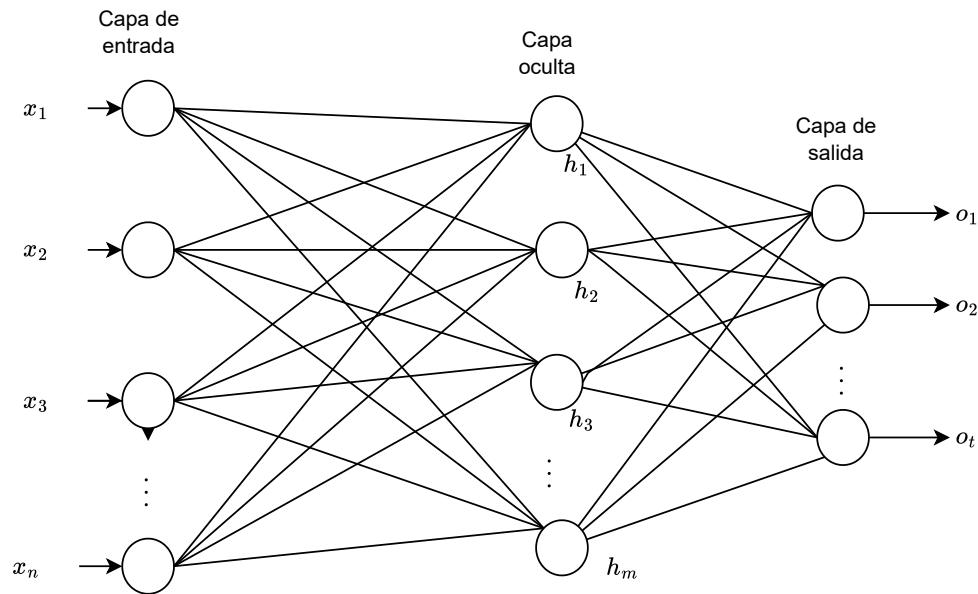


Figura 2.2: Arquitectura de una red neuronal artificial (RNA) de tres capas con alimentación hacia adelante (*feed-forward*). El diagrama muestra el flujo de información desde la capa de entrada (que recibe los datos originales), a través de una o más capas ocultas (que procesan y transforman la información mediante representaciones intermedias), hasta la capa de salida (que produce la respuesta final). Cada conexión entre neuronas tiene un peso asociado que se ajusta durante el entrenamiento mediante el algoritmo de retropropagación. Las capas ocultas permiten que la red aprenda representaciones jerárquicas y abstractas del problema, donde cada capa sucesiva captura patrones de mayor complejidad, habilitando así la capacidad de la red para modelar relaciones no lineales complejas entre los datos de entrada y salida.

entradas (definida como  $S$  en la Ecuación 2.2), generando una salida en un rango acotado, típicamente  $[0, 1]$  o  $[-1, 1]$ . Entre las funciones más utilizadas se encuentran *ReLU*, *sigmoide* y *tanh*, aunque otras como la gaussiana o lineal pueden ser útiles según el problema. En este trabajo se emplean *ReLU* y *sigmoide*, que han demostrado eficacia en distintas sub tareas; *ReLU*, definida como  $f(S) = \max(0, S)$ , destaca por su simplicidad y efectividad, mientras que la función sigmoide estándar se define como  $f(S) = \frac{1}{1 + \exp(-S)}$ , donde  $S$  es la suma



ponderada de las entradas (Ecuación 2.2), produciendo valores en el rango  $[0, 1]$ . La función sigmoide simétrica (también conocida como tangente hiperbólica o *tanh*) se define como  $f(S) = \tanh(S) = \frac{\exp(S) - \exp(-S)}{\exp(S) + \exp(-S)} = 2\sigma(2S) - 1$ , donde  $\sigma$  es la función sigmoide estándar, produciendo valores en el rango  $[-1, 1]$ . Estas funciones permiten controlar la no linealidad y manejar diferentes tipos de datos, contribuyendo al rendimiento del modelo.

### 2.3.5. Entrenamiento de RNAs

El entrenamiento de una red neuronal consiste en ajustar sus pesos para minimizar la diferencia entre la salida deseada y la generada por el modelo. Inicialmente, los pesos se asignan con valores pequeños y la información se propaga capa por capa hasta la salida, donde se calcula el error comparando el resultado obtenido con el esperado. Este error se utiliza para ajustar los pesos mediante el algoritmo de retropropagación (*backpropagation*), propuesto por Paul Werbos en 1974 [Werbos74], que permite propagar el error desde la capa de salida hacia las capas ocultas y actualizar los pesos usando el gradiente de la función de pérdida. En la Figura 2.3 se muestra el proceso de retropropagación. El proceso se realiza en tres fases principales: la propagación hacia adelante, donde se calcula la salida de cada neurona aplicando la función de activación sobre la suma ponderada de entradas; la propagación hacia atrás del error, donde se evalúa la discrepancia mediante funciones de pérdida como el error cuadrático medio para regresión o la entropía cruzada combinada con softmax para clasificación multiclase; y el cálculo del gradiente respecto a los pesos, aplicando la regla de la cadena, que permite ajustar los parámetros de manera eficiente. Este ciclo de propagación, cálculo de error y ajuste de pesos se repite durante múltiples iteraciones o épocas hasta que la red alcanza un rendimiento satisfactorio.

La elección de la función de pérdida en redes neuronales depende del tipo de problema a resolver (tabla 2.1). Para problemas de regresión, donde la salida es un valor continuo, se utiliza comúnmente la pérdida cuadrática media (*MSE*), que penaliza linealmente la diferencia entre la salida esperada y la predicha, aunque su uso en clasificación puede ser ineficiente, especialmente con activaciones como softmax. Para problemas de clasificación, la entropía cruzada (Cross-Entropy) es más adecuada, ya que compara la distribución predicha con la real codificada en *one-hot*. La codificación one-hot representa cada clase como un

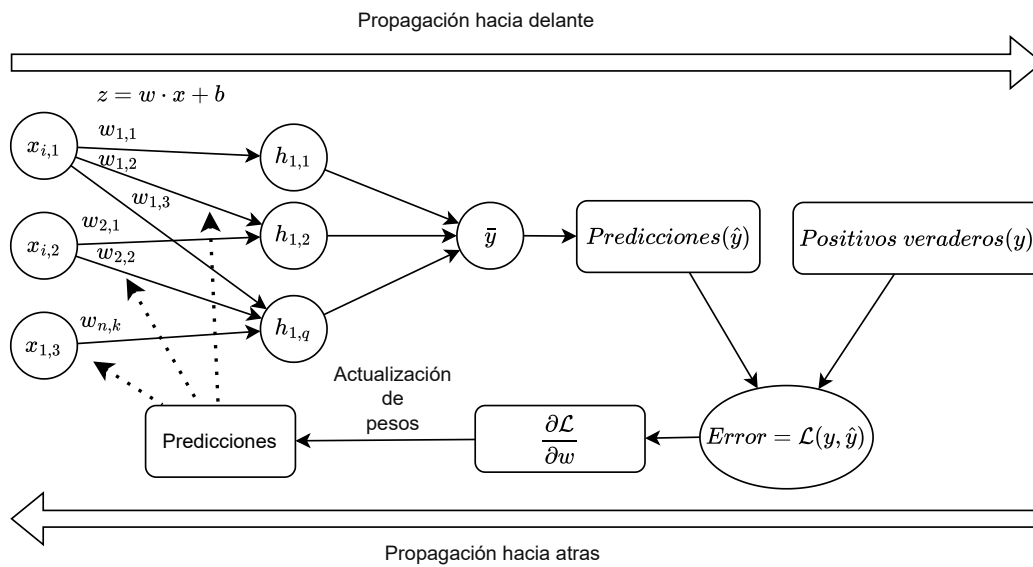


Figura 2.3: Proceso de entrenamiento mediante el algoritmo de retropropagación (*backpropagation*) en una red neuronal. El diagrama ilustra el ciclo completo de aprendizaje: (1) Propagación hacia adelante (*forward pass*), donde los datos de entrada fluyen a través de las capas de la red, cada neurona calcula su salida aplicando la función de activación a la suma ponderada de sus entradas; (2) Cálculo del error en la capa de salida comparando la salida predicha con la salida deseada utilizando una función de pérdida (como error cuadrático medio o entropía cruzada); (3) Propagación hacia atrás (*backward pass*), donde el error se propaga desde la capa de salida hacia las capas ocultas, calculando el gradiente de la función de pérdida respecto a cada peso mediante la regla de la cadena; (4) Actualización de pesos, donde cada peso se ajusta en dirección opuesta al gradiente usando un algoritmo de optimización (como descenso por gradiente o *Adam*), minimizando así el error y mejorando el rendimiento de la red. Este proceso se repite iterativamente hasta que la red converge hacia una configuración óptima de pesos.

vector donde solo la posición correspondiente a la clase verdadera tiene valor 1, mientras que todas las demás posiciones tienen valor 0. Por ejemplo, en un problema de 3 clases, la clase 0 se representa como  $[1,0,0]$ , la clase 1 como  $[0,1,0]$  y la clase 2 como  $[0,0,1]$ . Esta función

de pérdida penaliza con mayor intensidad las predicciones incorrectas con alta confianza, lo que favorece una convergencia más rápida y estable. En cuanto al entrenamiento mediante retropropagación, existen dos enfoques principales: el entrenamiento por lotes (*batch learning*), que calcula gradientes usando todo el conjunto de datos y actualiza los pesos una vez por época, proporcionando estabilidad a costa de mayor memoria; y el entrenamiento incremental (*online learning*), que actualiza los pesos tras cada instancia, permitiendo convergencia rápida pero con mayor variabilidad. Además, diversas variantes del descenso por gradiente, como *SGD*, *Adam* [Kingma14] y *RMSprop*, han sido desarrolladas para mejorar la eficiencia y estabilidad del entrenamiento.

Tabla 2.1: Comparación entre Funciones de Pérdida *MSE* y Entropía Cruzada

Característica	Pérdida Cuadrática Media ( <i>MSE</i> )	Entropía Cruzada ( <i>CE</i> )
Tipo de problema	Regresión	Clasificación binaria o multiclase
Salida esperada	Valores continuos	Distribución one-hot
Función de activación	ReLU, tanh	Softmax o sigmoide
Penalización de errores	Penaliza linealmente	Penaliza con mayor intensidad
Convergencia	Más lenta y menos estable	Más rápida y eficiente
Fórmula	$\mathcal{L}_{MSE} = \frac{1}{n} \sum (y_i - \hat{y}_i)^2$	$\mathcal{L}_{CE} = - \sum y_i \log(\hat{y}_i)$

Las Redes Neuronales Artificiales (RNAs) constituyen una herramienta fundamental para modelar sistemas complejos gracias a su capacidad de aprendizaje, generalización y versatilidad, lo que las hace especialmente útiles en tareas de procesamiento de lenguaje natural (PLN), como la detección de discriminación y la identificación de discursos discriminatorios, incluido el sexismo. Comprender las RNAs es crucial para este trabajo, ya que modelos de lenguaje modernos como *RoBERTa* se basan en arquitecturas neuronales profundas, y los principios de aprendizaje y optimización de estas redes, especialmente el

algoritmo de retropropagación, proporcionan la base teórica para entender cómo los algoritmos genéticos pueden mejorar su rendimiento. Sin embargo, lograr un desempeño óptimo requiere considerar cuidadosamente aspectos del diseño de la red, incluyendo el número de capas ocultas, neuronas por capa, funciones de activación y algoritmos de entrenamiento. Aun definido el diseño, ajustar los hiperparámetros representa un desafío significativo; este trabajo se centra precisamente en facilitar la optimización del diseño y entrenamiento de RNAs, dado que la falta de guías claras ha limitado históricamente su uso a expertos en el área.

## 2.4. Algoritmos Genéticos (AGs)

Los *Algoritmos Genéticos (AGs)* son técnicas de optimización estocástica inspiradas en la evolución biológica, particularmente en la selección natural propuesta por Darwin, formalizadas por John Holland en la década de 1970 [Holland75] y popularizadas por David E. Goldberg [Goldberg89] en aplicaciones de ingeniería. Estos algoritmos simulan la evolución de una población de soluciones candidatas mediante operadores genéticos como selección, cruce (*crossover*) y mutación, con el objetivo de explorar y explotar eficientemente espacios de búsqueda complejos. Cada solución se representa como un *individuo* o  *cromosoma*, codificado típicamente como una cadena binaria, aunque también pueden usarse representaciones en punto flotante, enteros o estructuras más complejas.

La población inicial se genera como un conjunto de individuos  $\mathcal{P}^{(0)} = \{x_1, x_2, \dots, x_N\}$ , cada uno representando una solución potencial en el espacio de búsqueda. En general, se codifica cada individuo como:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{il}) \quad \text{con } x_{ij} \in \{0, 1\} \text{ o } \mathbb{R} \quad (2.4)$$

Cada individuo de la población es evaluado mediante una función de aptitud  $f(x_i)$ , la cual cuantifica la calidad de la solución en relación con el objetivo de optimización. En el caso de problemas orientados a la minimización del error  $\mathcal{L}$ , la función de aptitud se define como:

$$f(x_i) = \frac{1}{1 + \mathcal{L}(x_i)} \quad (2.5)$$

De esta forma, valores más bajos de  $\mathcal{L}(x_i)$  se traducen en una mayor aptitud del individuo. El proceso de selección determina qué soluciones serán candidatas para participar en la fase de reproducción. Para ello, se emplean diferentes estrategias, entre las cuales destacan: (i) el método de ruleta, donde la probabilidad de selección es proporcional al valor de aptitud; (ii) el método de torneo, en el cual se elige al mejor individuo entre un subconjunto aleatorio de tamaño  $k$ :

$$x_{\text{win}} = \arg \max_{x \in T} f(x), \quad \text{donde } T \subseteq \mathcal{P}, |T| = k \quad (2.6)$$

y (iii) el elitismo, técnica que garantiza la preservación de los mejores individuos de la población para la siguiente generación.

El cruce (*crossover*) combina información de dos padres para generar nuevos individuos, intercambiando componentes con probabilidad 0.5 para cada gen:

$$x_j^{\text{hijo}} = \begin{cases} x_j^1, & \text{con probabilidad 0.5} \\ x_j^2, & \text{con probabilidad 0.5} \end{cases} \quad \forall j \in \{1, 2, \dots, l\} \quad (2.7)$$

Mientras que la mutación altera aleatoriamente uno o más genes con baja probabilidad  $p_m$ , promoviendo diversidad genética:

$$x'_j = x_j + \mathcal{N}(0, \sigma^2), \quad \text{con probabilidad } p_m \quad (2.8)$$

donde  $\mathcal{N}(0, \sigma^2)$  es una distribución normal con media 0 y varianza  $\sigma^2$ . Finalmente, la evolución se realiza mediante el reemplazo total o parcial de la población, formando  $\mathcal{P}^{(t+1)}$ , y el ciclo de selección, cruce y mutación se repite durante varias generaciones hasta alcanzar un criterio de parada o un número máximo de iteraciones.

Los AGs son especialmente útiles en problemas donde el espacio de búsqueda es complejo, no lineal, discontinuo o de múltiples óptimos locales. El flujo general se muestra en la Figura 2.4 y el algoritmo en 1.

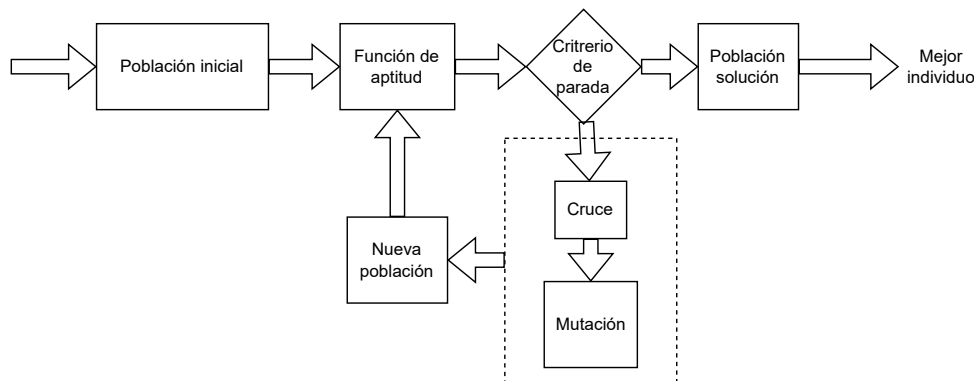


Figura 2.4: Flujo general de un algoritmo genético aplicado a la optimización de hiperparámetros. El diagrama muestra el ciclo evolutivo completo: (1) Inicialización de la población con individuos aleatorios codificando diferentes configuraciones de hiperparámetros; (2) Evaluación de cada individuo mediante la función de aptitud, que mide el rendimiento del modelo con esa configuración de hiperparámetros; (3) Selección de padres basándose en la aptitud, utilizando métodos como torneo o ruleta para favorecer individuos con mejor rendimiento; (4) Aplicación de operadores genéticos: cruza (*crossover*) que combina información de dos padres para generar descendencia, y mutación que introduce variaciones aleatorias para mantener diversidad; (5) Formación de nueva generación combinando individuos seleccionados y descendencia generada; (6) Reemplazo de la población anterior con la nueva generación; (7) Verificación de condición de parada (número máximo de generaciones o convergencia); si no se cumple, el ciclo se repite desde la evaluación. Este proceso iterativo permite explorar eficientemente el espacio de hiperparámetros, convergiendo hacia configuraciones óptimas que maximizan el rendimiento del modelo.

---

**Algorithm 1:** Algoritmo Genético (AG)

---

**Input:** Tamaño de la población  $N$ , número de generaciones  $G$ **Output:** Mejor individuo encontrado**Inicializar** la población  $P$  con  $N$  individuos generados aleatoriamente;Evaluar la función de aptitud de cada individuo en  $P$ ;**while** *No se cumpla la condición de parada (número de generaciones o criterio de convergencia)* **do**    **Seleccionar** un subconjunto de individuos como padres basándose en la función de aptitud;    **Aplicar operadores genéticos**;    **Evaluar** la función de aptitud de la nueva población;    **Actualizar** la población  $P$  con la nueva generación;Retornar el mejor individuo encontrado;

---

Los *Algoritmos Genéticos (AGs)* han demostrado ser herramientas efectivas para resolver problemas de optimización en el ámbito del *Machine Learning (ML)*, especialmente en escenarios donde el espacio de búsqueda es complejo, no diferenciable o altamente dimensional. Su naturaleza poblacional, estocástica y libre de derivadas les permite explorar eficientemente soluciones en funciones objetivo que presentan múltiples óptimos locales o comportamientos no suaves [Goldberg89, Sivanandam08]. Entre sus ventajas destacan la capacidad de exploración global, que permite escapar de óptimos locales accediendo a regiones del espacio de búsqueda inaccesibles para métodos basados en gradientes; la flexibilidad, al no requerir funciones diferenciables y permitir optimizar funciones complejas o discontinuas; la paralelización, que facilita la evaluación simultánea de múltiples soluciones aprovechando recursos distribuidos; y la adaptabilidad, que permite el ajuste dinámico de parámetros según las características del problema. Sus aplicaciones más relevantes incluyen la selección de características para identificar variables importantes y mejorar el rendimiento del modelo, la optimización automática de hiperparámetros, la búsqueda de arquitecturas neuronales explorando diferentes topologías y conexiones, y la optimización de funciones objetivo complejas, equilibrando múltiples criterios de evaluación de manera simultánea.

El uso de algoritmos genéticos (AGs) en tareas de Procesamiento de Lenguaje Natural (PLN) ha emergido como una estrategia poderosa para optimizar modelos y selec-

cionar características relevantes en espacios de búsqueda altamente complejos. Esto resulta particularmente útil en modelos de arquitectura profunda como *RoBERTa* o *BERT*, donde la cantidad de hiperparámetros y configuraciones posibles es masiva y no lineal. Entre sus principales ventajas en PLN se destacan la capacidad de exploración sin necesidad de derivadas, la robustez en espacios de alta dimensionalidad que evita quedar atrapado en regiones limitadas, la flexibilidad de codificación que permite ajustar desde hiperparámetros hasta arquitecturas completas, y la habilidad para escapar de óptimos locales, encontrando configuraciones globalmente óptimas. Sin embargo, los AGs enfrentan desafíos específicos en este dominio, como el alto costo computacional derivado de la evaluación de modelos de lenguaje completos o parciales, la limitada escalabilidad en conjunto de datos de gran tamaño y la complejidad en el diseño de funciones de aptitud que deben balancear múltiples métricas de rendimiento simultáneamente.

## 2.5. Brecha Actual y Oportunidades

Los enfoques tradicionales para la detección automática de contenido discriminatorio en texto suelen apoyarse en representaciones vectoriales estáticas como *Bag-of-Words* (*BoW*) [Harris54, Salton88], *TF-IDF* [Sparck Jones72, Salton88] o *word embeddings* pre-entrenados sin contexto [Mikolov13, Pennington14], combinados con clasificadores lineales como máquinas de vectores de soporte (*SVM*) o regresión logística [Mohammad17]. Sin embargo, estas técnicas presentan limitaciones importantes, como la incapacidad de capturar el significado contextual dependiente de la posición y la dificultad para modelar relaciones semánticas no lineales complejas. La integración de Algoritmos Genéticos (AGs) en la optimización de modelos de lenguaje profundo, específicamente *RoBERTa*, constituye una estrategia poderosa para superar estos desafíos. Los AGs permiten explorar espacios de búsqueda altamente complejos sin requerir derivadas ni conocimiento previo sobre la función de aptitud, facilitando la optimización de hiperparámetros críticos como la tasa de aprendizaje, tamaño de lote de entrenamiento, tasa de abandono, así como la configuración de capas adicionales y funciones de activación en la arquitectura de *RoBERTa*. La definición de una función de aptitud especializada combina métricas de rendimiento con penalizacio-



---

nes por complejidad, considerando el equilibrio entre clases, el tiempo de entrenamiento y el tamaño del modelo, lo que permite adaptar el modelo de manera eficiente a corpus sociales dinámicos y con alta ambigüedad semántica. Este enfoque asegura que *RoBERTa* optimizado mediante AGs pueda generalizar mejor, adaptándose a patrones cambiantes en el lenguaje y mejorando la detección de contenido discriminatorio, superando así las limitaciones de los métodos tradicionales y ofreciendo un marco sistemático para la optimización evolutiva de modelos de lenguaje profundo.

## 2.6. Comentarios Finales

Este capítulo presentó los fundamentos teóricos y el estado del arte que sustentan la optimización de modelos de lenguaje mediante algoritmos genéticos para la detección de sexismo en redes sociales. Se revisaron los principales enfoques para identificar contenido discriminatorio, desde métodos subjetivos iniciales hasta las arquitecturas modernas de procesamiento del lenguaje natural.

La evolución de los modelos de lenguaje, desde los n-gramas hasta los *transformers*, evidenció la importancia de capturar contexto semántico profundo y relaciones no lineales. Los mecanismos de atención permiten detectar patrones discriminatorios y microagresiones que los métodos tradicionales no identifican.

Las redes neuronales artificiales constituyen la base computacional de estos modelos, gracias a su capacidad de aprendizaje y generalización. Los algoritmos de entrenamiento y funciones de activación son elementos esenciales para el aprendizaje efectivo de estas arquitecturas.

Los algoritmos genéticos se consolidan como herramientas eficaces para optimizar hiperparámetros en espacios de búsqueda complejos y no diferenciables. Su naturaleza poblacional y estocástica facilita escapar de óptimos locales y hallar configuraciones globalmente óptimas.

Finalmente, la brecha entre los métodos tradicionales y las necesidades actuales en la detección de contenido discriminatorio respalda una estrategia evolutiva que combine modelos de lenguaje modernos con algoritmos genéticos para optimizar hiperparámetros.



## Capítulo 3

# Metodología

Este capítulo presenta la metodología completa desarrollada para la optimización de modelos de lenguaje mediante algoritmos genéticos aplicada a la detección de sexismo en redes sociales. Se describe el conjunto de datos *EXIST* 2025 [Basile25] utilizado para la evaluación, el diseño de la estrategia de optimización evolutiva que integra algoritmos genéticos con el modelo *RoBERTa*, las métricas de evaluación seleccionadas para medir el rendimiento en las tres subtareas del problema, y la implementación técnica que asegura la reproducibilidad de los experimentos. La metodología propuesta establece un marco sistemático para la optimización automática de hiperparámetros en modelos de lenguaje profundo para tareas de detección de contenido sensible.

### 3.1. Descripción del Conjunto de Datos

El conjunto de datos utilizado en este estudio corresponde a la Tarea 1 del desafío internacional *CLEF-2025*, denominado “*EXIST: Sexism Identification and Characterization in Tweets*” [Basile25]. Este conjunto de datos fue construido a partir de publicaciones en la red social X, anotadas manualmente por múltiples evaluadores, con el objetivo de identificar y caracterizar expresiones sexistas en distintos niveles y matices discursivos. El Oxford English Dictionary define el sexismo como: “*prejudice, stereotyping or discrimination, typically against women, on the basis of sex*” [Oxford English Dictionary24]. En términos generales, los comportamientos y discursos sexistas subestiman el papel de las mujeres. Las

desigualdades y formas de discriminación estructural contra las mujeres, aún presentes en las sociedades actuales, tienden a replicarse y amplificarse en entornos digitales.

No obstante, detectar sexismo en línea es particularmente desafiante, ya que puede manifestarse de manera diversa y sutil. Por ejemplo:

- Puede sonar “**amigable**”, como en: *“Mi hijo, ayer dijo: “quiero más a mi mamá”. Yo inmediatamente me puse triste. Después me dijo, a las mujeres hay que amarlas un poquito más”*, lo cual perpetúa la idea de inferioridad o debilidad.
- Puede presentarse como “**humorístico**” o banal: *“Jajajaja a las mujeres hay que quererlas antes de entenderlas. Buen día mi estimado Lobo!! Abrazote”*.
- Puede adquirir formas **abiertamente ofensivas y violentas**: *“Hace 10 años me grito perra malparida , estando yo embarazada y lo sigue haciendo aun , que tierno”*.

Incluso las formas más sutiles pueden resultar tan perjudiciales como las explícitas, afectando diversas dimensiones de la vida de las mujeres: relaciones familiares, oportunidades laborales, imagen sexual, aspiraciones de vida, entre otras.

En este contexto, el conjunto de datos *EXIST* 2025 contempla tres subtareas de clasificación:

1. **Tarea 1.1 - Detección de Sexismo:** Clasificación binaria (*sexista* vs. *no sexista*).
2. **Tarea 1.2 - Intención del mensaje:** Clasificación multiclase en tres categorías: *intencional*, *no intencional* y *reportado*.
3. **Tarea 1.3 - Categorías del Sexismo:** Clasificación multiclase en cinco categorías según la manifestación del prejuicio o discriminación.

Las anotaciones fueron realizadas por seis evaluadores por tweet, y se consolidaron utilizando voto mayoritario. Esta riqueza anotativa permite abordar el problema desde una perspectiva multitarea, aprovechando modelos de lenguaje como *RoBERTa* para abordar simultáneamente las tres dimensiones.

## 3.2. Estructura del Conjunto de Datos

El conjunto de datos *EXIST* 2025 presenta una estructura *JSON* bien definida donde cada instancia contiene información detallada sobre el tweet y sus anotaciones. Cada entrada del conjunto de datos incluye un identificador único (`id_EXIST`), el idioma del tweet (`lang`), y el contenido textual completo del tweet. La información de anotación es particularmente rica, incluyendo datos demográficos de los seis evaluadores que participaron en la anotación de cada instancia: género (`gender_annotators`), rango de edad (`age_annotators`), etnicidad (`ethnicities_annotators`), nivel de estudios (`study_levels_annotators`) y país de origen (`countries_annotators`).

Para cada una de las tres subtareas, el conjunto de datos proporciona las etiquetas individuales de los seis evaluadores. En la Tarea 1.1 (detección binaria de sexismo), las etiquetas pueden ser “*YES*” o “*NO*”, mientras que en la Tarea 1.2 (intención del mensaje) se utilizan las categorías “*DIRECT*”, “*JUDGEMENTAL*”, “*REPORTED*” o “-” para indicar ausencia de anotación. La Tarea 1.3 (categorización de sexismo) es la más compleja, ya que permite múltiples categorías simultáneas como “*STEREOTYPING-DOMINANCE*”, “*IDEOLOGICAL-INEQUALITY*”, “*OBJECTIFICATION*”, “*MISOGYNY-NON-SEXUAL-VIOLENCE*” y “*SEXUAL-VIOLENCE*”. Cada tweet también incluye información sobre el conjunto de datos al que pertenece (`split`), permitiendo una división adecuada entre entrenamiento, desarrollo y prueba.

La diversidad demográfica de los anotadores, que incluye diferentes géneros, edades, etnicidades y nacionalidades, contribuye a la robustez del conjunto de datos al capturar diferentes perspectivas culturales y sociales sobre lo que constituye contenido sexista. Esta estructura multicapa permite no solo la evaluación del rendimiento del modelo en las tareas principales, sino también el análisis de cómo diferentes grupos demográficos interpretan el contenido discriminatorio, proporcionando conocimientos valiosos para el desarrollo de sistemas más inclusivos y culturalmente sensibles.

### 3.3. Diseño de la Estrategia de Optimización

La elección de una estrategia de optimización adecuada es crucial para maximizar el desempeño de modelos de lenguaje profundo como *RoBERTa* en tareas complejas y sensibles, como la detección y clasificación de contenido discriminatorio en redes sociales. En este trabajo, se adoptó un enfoque basado en Algoritmos Genéticos (AGs) para el ajuste automático de hiperparámetros críticos del modelo multitarea. El modelo base seleccionado, *RoBERTa-base*, se adaptó mediante tres capas de clasificación independientes para abordar simultáneamente las tareas de detección binaria, clasificación de intención y categorización de contenido discriminatorio. Cada individuo en la población del AG codifica una combinación de hiperparámetros clave: la tasa de aprendizaje ( $lr$ ), que controla la magnitud de los ajustes de los pesos durante el entrenamiento; el tamaño de lote ( $b$ ), que define la cantidad de ejemplos procesados antes de actualizar los pesos; la tasa de abandono ( $d$ ), que regula la probabilidad de omitir neuronas durante el entrenamiento para evitar sobreajuste; los pesos de pérdida de cada subtarea ( $w_1, w_2, w_3$ ), que equilibran la importancia relativa de cada tarea en la función de aptitud; y el número de capas entrenables de *RoBERTa* ( $n_{layers}$ ), que controla cuántas de las 12 capas transformer del modelo base se actualizan durante el ajuste fino. La población inicial consta de 20 individuos que evolucionan durante 8 generaciones mediante selección por torneo ( $k = 3$ ), cruza uniforme de dos puntos y mutación gaussiana adaptativa, explorando eficientemente combinaciones óptimas en el espacio de búsqueda. La función de aptitud combina los F1-scores de las tres subtarear ponderados por  $w_1, w_2$  y  $w_3$ , permitiendo al AG maximizar simultáneamente el desempeño global del modelo. La tabla 3.1 resume los hiperparámetros optimizados y sus rangos explorados.

Tabla 3.1: Espacio de búsqueda de hiperparámetros para la optimización de *RoBERTa* mediante Algoritmos Genéticos

Hiperparámetro	Rango
Tasa de aprendizaje ( $lr$ )	$[1 \times 10^{-5}, 5 \times 10^{-5}]$
Tamaño de lote ( $b$ )	$\{16, 32\}$
Tasa de abandono ( $d$ )	$[0.1, 0.4]$
Peso subtarea 1 ( $w_1$ )	$[0.8, 1.2]$
Peso subtarea 2 ( $w_2$ )	$[1.0, 1.5]$
Peso subtarea 3 ( $w_3$ )	$[2.0, 2.5]$
Capas entrenables de <i>RoBERTa</i> ( $n_{layers}$ )	$\{3, 6, 12\}$

El proceso de evaluación de cada individuo dentro del Algoritmo Genético comienza con la configuración del modelo *RoBERTa*, en la cual se aplican directamente los hiperparámetros codificados en su representación genética. Esta fase inicial es importante, ya que define la estructura y los parámetros de entrenamiento condicionan el desempeño del modelo en las siguientes etapas. Una vez configurado, se procede a un entrenamiento parcial limitado a tres épocas, lo que permite acelerar el proceso sin perder fiabilidad en la estimación del valor de aptitud. Esta estrategia de entrenamiento rápido actúa como un mecanismo de muestreo que proporciona información suficiente sobre la capacidad del modelo para generalizar, al tiempo que reduce de manera significativa los costos computacionales asociados a un entrenamiento completo.

Después de el entrenamiento, se calculan las métricas de desempeño en el conjunto de validación, obteniendo medidas claras de la efectividad del modelo en cada subtarea. Cada una de las métricas no se consideran de manera aislada, sino que se integran mediante una función de aptitud ponderada, diseñada específicamente para balancear los tres objetivos simultáneos: detección binaria, clasificación de intención y categorización de sexismo. Los pesos asignados a cada subtarea permiten ajustar la relevancia relativa de cada una, garantizando un proceso de optimización más justo y representativo de las demandas reales del problema.

Finalmente, todos los resultados, incluyendo configuraciones, métricas parciales y valores de aptitud, son almacenados de forma sistemática para su análisis posterior. Esta trazabilidad no solo permite reproducir experimentos y validar hallazgos, sino que también facilita un análisis detallado del comportamiento evolutivo del algoritmo y de las tendencias de convergencia.

### 3.4. Métricas de Evaluación

La evaluación del modelo multitarea se realiza empleando métricas estándar en clasificación, con énfasis en medidas de precisión, recuperación y F1-score. Dado que las tres subtareas (sección 3.1) implican distintos tipos de salida (binaria, multiclase y multicategoría), se utilizan métricas adaptadas a cada contexto. El uso de métricas apropiadas es fundamental para evaluar la robustez del modelo, especialmente en escenarios con clases desbalanceadas como en la detección de sexismo [Pedregosa11]. En la tabla 3.2 se presenta un resumen de las métricas utilizadas por subtarea.

La evaluación del modelo multitarea se realiza mediante métricas estándar en clasificación, priorizando precisión, recuperación y F1-score, ya que cada subtarea implica un tipo distinto de salida (binaria, multiclase o multicategoría). La exactitud (*accuracy*) se emplea como medida global de aciertos, aunque resulta poco informativa en escenarios con clases desbalanceadas. Para esta limitación se consideran la precisión, que mide la proporción de verdaderos positivos entre las predicciones positivas, y la recuperación, que refleja la capacidad del modelo para identificar los casos relevantes; estas dos medidas se combinan en el F1-score, que balancea ambos aspectos. En el contexto multiclase se utilizan variantes agregadas como el F1 ponderado, que incorpora el soporte de cada clase, y el F1 macro, que promedia de manera uniforme las métricas de todas las clases. Además, el F1 micro se incluye como alternativa sensible al desbalance, ya que calcula métricas acumuladas en lugar de promedios por clase. Estas medidas permiten evaluar de manera integral la robustez del modelo frente a distintos escenarios, asegurando un análisis justo en tareas de detección de sexismo, donde la distribución de categorías suele ser desigual.



- **Accuracy Score:**

$$Accuracy = \frac{\text{Número de predicciones correctas}}{\text{Total de predicciones}} \quad (3.1)$$

- **Precision, Recall y F1-Score por clase  $i$ :**

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad Recall_i = \frac{TP_i}{TP_i + FN_i} \quad F1_i = 2 \cdot \frac{Precision_i \cdot Recall_i}{Precision_i + Recall_i} \quad (3.2)$$

- **F1 Ponderado (Weighted F1):**

El F1 ponderado considera la frecuencia de cada clase en el conjunto de datos mediante el concepto de soporte. El soporte de una clase  $i$  se define como el número total de instancias verdaderas de esa clase en el conjunto de datos:

$$\text{Soporte}_i = TP_i + FN_i \quad (3.3)$$

donde  $TP_i$  son los verdaderos positivos y  $FN_i$  son los falsos negativos para la clase  $i$ . El F1 ponderado se calcula como:

$$F1_{\text{weighted}} = \sum_{i=1}^N w_i \cdot F1_i \quad \text{donde} \quad w_i = \frac{\text{Soporte}_i}{\text{Total de muestras}} \quad (3.4)$$

- **F1 Micro y F1 Macro:** Estas métricas son especialmente útiles en tareas multiclase y multilabel.

$$F1_{\text{micro}} = \frac{2 \cdot \sum TP}{2 \cdot \sum TP + \sum FP + \sum FN} \quad F1_{\text{macro}} = \frac{1}{N} \sum_{i=1}^N F1_i \quad (3.5)$$

Tabla 3.2: Resumen de Métricas por Subtarea

Subtarea	Tipo de Clasificación	Métricas Utilizadas
1.1	Binaria (Sexismo sí/no)	<i>Accuracy Score</i> ( 3.1), <i>Precision</i> , <i>Recall</i> , <i>F1 Score</i> ( 3.2)
1.2	Multiclase (3 clases)	<i>F1 Weighted</i> ( 3.4), <i>F1 Macro</i> ( 3.5)
1.3	Multiclase (5 clases)	<i>F1 Micro and F1 Macro</i> ( 3.5)

## 3.5. Implementación

### 3.5.1. Entorno de Desarrollo y Herramientas

La implementación del modelo optimizado se llevó a cabo en un entorno de cómputo con soporte para procesamiento en *GPU*, utilizando *Python 3.11* y *CUDA Toolkit 11.8*. Este entorno permitió acelerar los procesos de entrenamiento, especialmente debido a la naturaleza intensiva en cómputo de los modelos de lenguaje como *RoBERTa*.

Las principales herramientas empleadas en esta fase fueron:

- ***Transformer de HuggingFace*** [Wolf20]: Proporciona acceso a modelos preentrenados como *RoBERTa-base*, facilitando la tokenización, el entrenamiento y la evaluación sobre tareas multitarea.
- ***PyTorch*** [Paszke19]: Usado para definir la arquitectura del modelo, gestionar el flujo de tensores y entrenar los clasificadores multitarea.
- ***DEAP (Distributed Evolutionary Algorithms)*** [Fortin12]: Biblioteca empleada para implementar algoritmos genéticos (AGs) personalizados, permitiendo definir operadores de selección, cruce, mutación y función de aptitud.
- ***Scikit-learn*** [Pedregosa11]: Utilizada para el cálculo de métricas de evaluación (*Precision*, *Recall*, *F1*) y análisis estadístico de resultados.
- ***Matplotlib y Seaborn***: Usadas para la visualización del rendimiento del modelo durante el entrenamiento y el análisis exploratorio de datos.

El entorno fue configurado en un equipo con las siguientes especificaciones: *GPU NVIDIA RTX 3000 (6GB)*, *32 GB de RAM*, y sistema operativo Windows 11 Pro.

### 3.5.2. Entrenamiento y evaluación

En cada una de las subtareas se utilizó la función de activación *Softmax*, ya que permite convertir las salidas lineales de las capas densas en distribuciones de probabilidad normalizadas sobre las clases posibles. En la subtarea binaria (sexismo vs. no sexismo) se empleó un *Softmax* de dos dimensiones en lugar de una sigmoide, ya que esto asegura simetría en la representación de ambas clases y mantiene consistencia con el resto de las subtareas multiclase. En la segunda subtarea, correspondiente a la clasificación de intención con tres categorías, el *Softmax* de tres dimensiones facilita asignar una probabilidad explícita a cada clase mutuamente excluyente, lo que resulta apropiado para problemas donde un mismo enunciado no puede pertenecer a más de una categoría simultáneamente. Finalmente, en la tercera subtarea, que aborda la categorización del tipo de sexismo en cinco clases, se recurrió al *Softmax* de cinco dimensiones por las mismas razones: garantizar exclusividad entre clases y proporcionar interpretabilidad probabilística en la salida del modelo. De esta manera, el uso sistemático de *Softmax* en las tres cabeceras asegura coherencia entre tareas, estabilidad en el entrenamiento y comparabilidad directa entre las predicciones.

El proceso de entrenamiento y evaluación se estructuró en una rutina automatizada que permite integrar el algoritmo genético con la optimización del modelo multitarea *RoBERTa*. Las etapas clave se describen a continuación:

1. Carga y preprocesamiento del conjunto de datos consolidando las múltiples anotaciones por tweet, dividiendo el conjunto en entrenamiento (80 %) y validación (20 %). Los textos fueron convertidos a minúsculas, eliminando duplicados y caracteres especiales.
2. Tokenización multitarea, cada entrada con codificación paralela para las tres subtareas (sección 3.1).
3. Construcción del modelo multitarea usando *RoBERTa* (sección 3.2.1), con una arquitec-

tura extendida que incluye tres cabeceras independientes:

$$RoBERTa(x) \rightarrow \begin{cases} f_{\text{binaria}}(h) = \text{Softmax}_2(W_1h + b_1) \\ f_{\text{multi3}}(h) = \text{Softmax}_3(W_2h + b_2) \\ f_{\text{multi5}}(h) = \text{Softmax}_5(W_3h + b_3) \end{cases}$$

**Donde:**

- $RoBERTa(x)$  representa la salida del codificador  $RoBERTa$  aplicada a la entrada  $x$ .
- $h_{[\text{CLS}]}$  corresponde al vector de representación asociado al token especial  $[\text{CLS}]$ , que resume el significado semántico del texto completo.
- Cada función  $f$  es una capa densa seguida de una activación  $Softmax$  diseñada específicamente para cada subtarea:
  - $f_{\text{binaria}}$ : Subtarea 1.1 – Clasificación binaria (sexismo vs. no sexismo).
  - $f_{\text{multi3}}$ : Subtarea 1.2 – Clasificación multiclase de intención (3 clases).
  - $f_{\text{multi5}}$ : Subtarea 1.3 – Clasificación multiclase de tipo de sexismo (5 clases).
- Cada una de las capas densas es parametrizada por un conjunto de pesos  $W_i$  y términos de sesgo  $b_i$ , aprendidos durante el entrenamiento:

$$f_{\text{tarea}}(h) = \text{Softmax}_k(W_ih + b_i)$$

donde  $k$  indica el número de clases de la tarea correspondiente.

4. Optimización con algoritmo genético para encontrar la mejor configuración de hiperparámetros mencionados en la sección 3.2.3.
5. Entrenamiento por individuo dada una configuración completa del modelo multitarea durante 3 épocas, almacenando los resultados y métricas.
6. Evaluación y selección del mejor individuo sobre todas las generaciones, se selecciona el individuo con mejor valor de aptitud para el entrenamiento final del modelo.

### 3.6. Comentarios Finales

Este capítulo presentó la metodología completa para la optimización de modelos de lenguaje mediante algoritmos genéticos, aplicada a la detección de sexismo en redes sociales. La estrategia se basó en el conjunto de datos EXIST 2025, que ofrece un marco robusto para evaluar sistemas de detección de contenido discriminatorio en tres subtareas: detección binaria, clasificación de intención y categorización de tipos de sexismo.

La integración sistemática de algoritmos genéticos con modelos de lenguaje profundo como *RoBERTa* constituye una contribución metodológica relevante. La codificación de hiperparámetros en individuos genéticos permite una exploración eficiente del espacio de búsqueda, considerando tasa de aprendizaje, tamaño de lote, tasa de abandono, pesos de pérdida y número de capas entrenables.

La función de aptitud ponderada equilibra las tres subtareas simultáneamente, evitando la dominancia de una sobre las demás y garantizando un desempeño uniforme. Esta aproximación multitarea resulta especialmente útil ante la complejidad multidimensional del fenómeno del sexismo.

Las métricas de evaluación utilizadas —*Accuracy*, *Precision*, *Recall* y *F1-Score* en sus variantes macro, micro y ponderada— ofrecen una valoración integral del modelo, atendiendo las particularidades y el desbalance natural del conjunto de datos.

La implementación, basada en *PyTorch*, *HuggingFace Transformers* y *DEAP*, asegura reproducibilidad y escalabilidad. El entrenamiento automatizado con tres épocas por evaluación evolutiva logra un equilibrio adecuado entre eficiencia computacional y confiabilidad.

En conjunto, la metodología establece un marco sólido y reproducible para la optimización evolutiva de modelos de lenguaje en la detección de contenido sensible, y constituye una base consistente para investigaciones futuras en esta línea.



## Capítulo 4

# Resultados y Análisis de la Optimización con Algoritmos Genéticos

Este capítulo presenta los resultados experimentales obtenidos mediante la implementación de la estrategia de optimización evolutiva descrita en el capítulo 3. Los objetivos específicos son evaluar la efectividad de los Algoritmos Genéticos en la optimización de hiperparámetros para modelos *RoBERTa* multitarea, cuantificar el impacto de la optimización evolutiva en las métricas de rendimiento para cada subtarea del desafío *EXIST 2025*, analizar la convergencia y eficiencia computacional del proceso, y caracterizar la robustez y capacidad de generalización del modelo optimizado.

### 4.1. Análisis de Resultados

En esta sección se presenta un análisis exhaustivo del rendimiento del modelo optimizado con Algoritmos Genéticos, incluyendo el proceso de optimización evolutiva, la convergencia del algoritmo, y la comparación con métodos tradicionales de ajuste fino. Es importante destacar que la evaluación se realizó siguiendo la metodología estándar de tres corpus: el corpus de desarrollo se utilizó para la optimización evolutiva, mientras que el

corpus de prueba se reservó para la evaluación final, garantizando métricas no sesgadas del rendimiento real del modelo.

El proceso de optimización se ejecutó durante 8 generaciones, evaluando un total de 114 configuraciones diferentes. La evolución del valor de aptitud a lo largo de las generaciones muestra una convergencia efectiva hacia configuraciones óptimas. La Tabla 4.1 presenta un resumen de la evolución del valor de aptitud a través de las generaciones:

Tabla 4.1: Evolución del valor de aptitud del algoritmo genético a través de 8 generaciones, mostrando la distribución de valores de aptitud para cada una de las 114 configuraciones evaluadas. El gráfico ilustra cómo el algoritmo converge progresivamente hacia valores superiores, evidenciando que las generaciones posteriores (especialmente la Generación 8) alcanzan valores de aptitud más altos (hasta 2.2404), mientras que las generaciones iniciales presentan mayor dispersión y valores más bajos. La mejora gradual en los valores mínimos y máximos de aptitud a lo largo de las generaciones demuestra la capacidad del algoritmo genético para explorar eficientemente el espacio de hiperparámetros y converger hacia configuraciones óptimas.

Generación	Individuos Evaluados	Valor de Aptitud Promedio	Mejor Valor de Aptitud
0	20	2.0898	2.2072
1	17	2.0742	2.2166
2	12	2.0968	2.2094
3	10	2.0956	2.2218
4	14	2.1605	2.2244
5	11	2.1641	2.2115
6	18	2.1721	2.2228
7	18	2.1721	2.2228
8	2	2.1869	2.2404

La gráfica 4.1 muestra la evolución del valor de aptitud a través de las ocho generaciones del algoritmo genético, revelando un patrón de progreso claro y consistente. La optimización comienza en la Generación 0 con valores que oscilan entre 1.4737 (mínimo) y 2.2072 (máximo), estableciendo una base inicial prometedora. A medida que avanza el proceso evolutivo, se observan mejoras significativas en las Generaciones 1, 4, 5, 7 y 8, donde



la Generación 8 alcanza el mejor resultado general de 2.2404. El patrón general muestra una convergencia hacia valores superiores a 2.2, con una estabilidad creciente después de la Generación 2, indicando que el algoritmo está efectivamente explorando y explotando el espacio de búsqueda de hiperparámetros de manera eficiente.

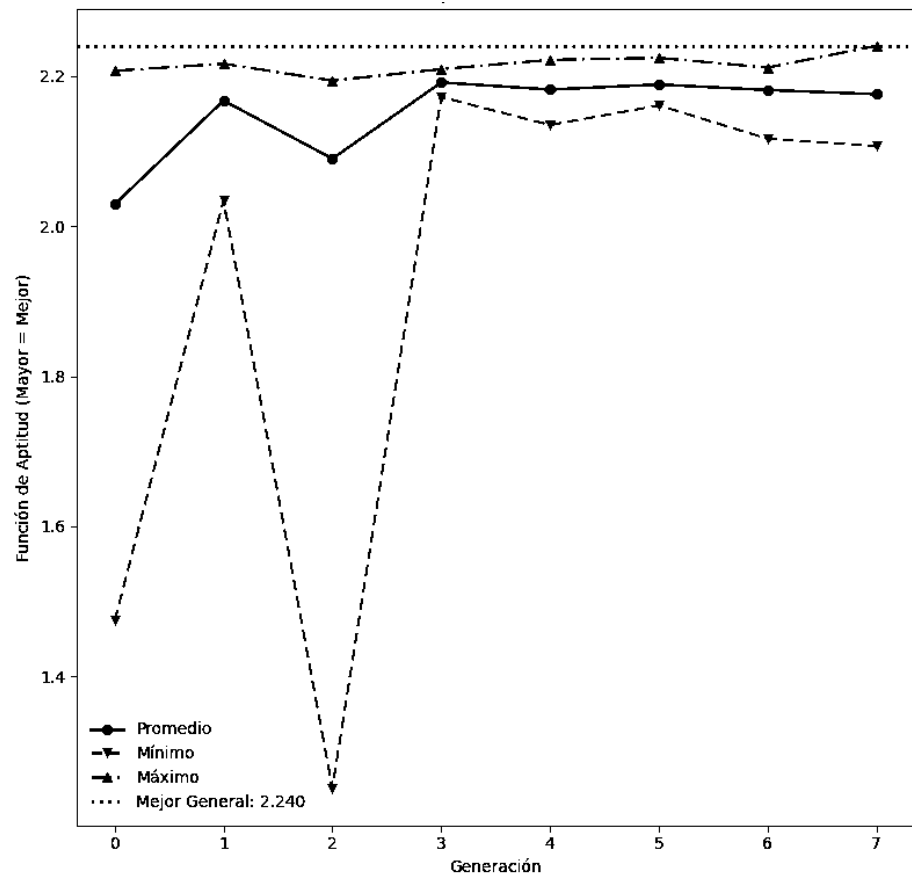


Figura 4.1: Evaluación de la función de aptitud durante 114 ejecuciones.

La gráfica 4.2 presenta una comparación directa entre el valor promedio y el mejor valor de aptitud para cada generación, proporcionando información valiosa sobre la consistencia del rendimiento. La Generación 3 destaca como la más estable, mostrando una diferencia mínima de solo 0.0178 entre su valor promedio (2.1916) y su mejor valor (2.2094), lo que indica una convergencia muy consistente. En contraste, la Generación 7 presenta una diferencia significativa (0.0507) entre su promedio (2.1721) y su mejor valor (2.2228), sugiriendo alta variabilidad pero buena capacidad de exploración. La Generación 8 alcanza el

mejor rendimiento absoluto con 2.2404, mostrando una diferencia moderada (0.0535) entre su promedio (2.1869) y su mejor valor. La tendencia general muestra que las diferencias entre valores promedio y mejores se reducen progresivamente después de la Generación 2, indicando una mejora en la estabilidad y consistencia del algoritmo genético a lo largo de las generaciones.

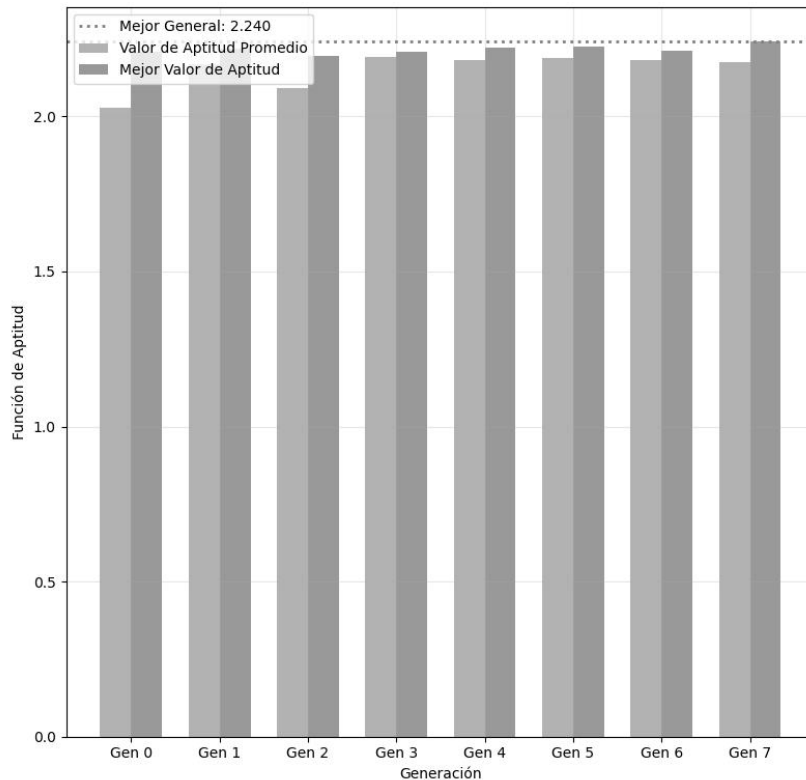


Figura 4.2: Comparación del valor promedio y el mejor valor de aptitud por generación del algoritmo genético. El gráfico muestra cómo ambas métricas evolucionan a lo largo de las 8 generaciones, revelando la convergencia del algoritmo y la estabilidad de la población. La proximidad entre el valor promedio y el mejor valor en cada generación indica la consistencia del rendimiento, mientras que la diferencia entre ambos refleja la diversidad de la población. La Generación 8 alcanza el mejor rendimiento absoluto (2.2404), y la tendencia general muestra una reducción progresiva en la diferencia entre promedio y mejor valor después de la Generación 2, indicando una mejora en la estabilidad y convergencia del algoritmo genético.

La gráfica 4.3 analiza la estabilidad de cada generación mediante el coeficiente de variación, donde valores más bajos indican mayor estabilidad. La Generación 3 emerge como la más estable de todas con un CV de 0.006, demostrando una consistencia excepcional en sus resultados. La Generación 1 también muestra excelente estabilidad con un CV de 0.021, mientras que la Generación 2 presenta la menor estabilidad con un CV de 0.127, principalmente debido a la presencia de un outlier bajo (1.2506) que aumenta significativamente la variabilidad. La tendencia general revela una mejora sustancial en la estabilidad después de la Generación 2, con todas las generaciones posteriores manteniendo CV por debajo de 0.021, lo que sugiere que el algoritmo ha alcanzado un estado de madurez donde puede producir resultados consistentes y confiables. Esta estabilidad mejorada es crucial para la implementación práctica de los hiperparámetros optimizados, ya que garantiza que el rendimiento sea reproducible y predecible.

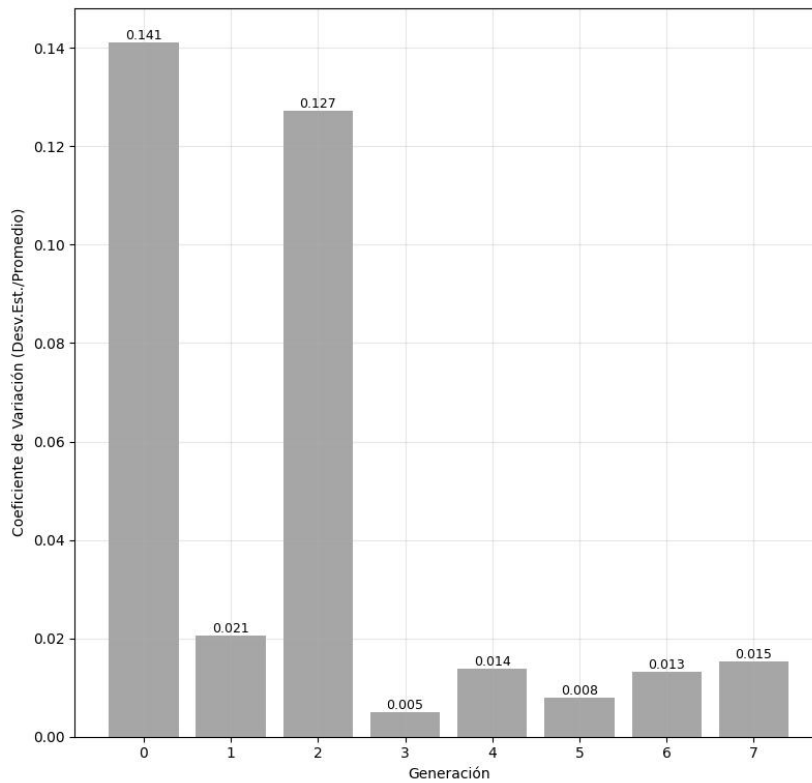


Figura 4.3: Análisis de estabilidad del algoritmo genético mediante el coeficiente de variación (CV) para cada generación. El gráfico muestra cómo el CV evoluciona a lo largo de las 8 generaciones, donde valores más bajos indican mayor estabilidad y consistencia en los resultados. La Generación 3 presenta el menor CV (0.006), demostrando la mayor estabilidad, mientras que la Generación 2 muestra el mayor CV (0.127) debido a la presencia de outliers. La tendencia general revela una mejora sustancial en la estabilidad después de la Generación 2, con todas las generaciones posteriores manteniendo CV por debajo de 0.021, lo que sugiere que el algoritmo ha alcanzado un estado de madurez donde produce resultados reproducibles y confiables. Esta estabilidad mejorada es crucial para garantizar que los hiperparámetros optimizados sean consistentes en implementaciones prácticas.

El proceso de optimización evolutiva demostró una eficiencia computacional notable, evaluando 114 configuraciones para encontrar la mejor configuración. Esta eficiencia se explica por la capacidad del algoritmo genético para realizar una exploración inteligente del espacio de búsqueda, concentrándose en regiones prometedoras a partir del historial de

---

valores de aptitud de generaciones anteriores. Asimismo, los operadores genéticos muestran una adaptación dinámica, ajustándose automáticamente según la diversidad de la población y la convergencia observada. Finalmente, el algoritmo mantiene un adecuado balance entre exploración y explotación, preservando la diversidad genética mientras avanza de manera consistente hacia soluciones óptimas.

La estrategia evolutiva adoptada ofrece ventajas sustanciales respecto a métodos tradicionales de búsqueda de hiperparámetros. En primer lugar, la exploración eficiente del espacio de búsqueda permite identificar configuraciones prometedoras evaluando únicamente 114 individuos, lo que representa un ahorro considerable en tiempo y recursos frente a técnicas de búsqueda exhaustiva o aleatoria. En segundo lugar, la adaptabilidad de los operadores genéticos asegura que el proceso de optimización no permanezca estático, sino que se ajuste dinámicamente en función del progreso alcanzado, favoreciendo una búsqueda cada vez más dirigida hacia regiones prometedoras del espacio. Asimismo, el diseño de la función de aptitud combinada evita la dominancia de una subtarea sobre las demás, garantizando un equilibrio en la optimización multitarea. Finalmente, el carácter estocástico del algoritmo dota al proceso de una capacidad inherente para escapar de óptimos locales, reduciendo el riesgo de convergencia prematura en soluciones subóptimas y aumentando las probabilidades de encontrar configuraciones de alta calidad dentro del espacio de búsqueda definido.

La Figura 4.1 muestra la evolución del valor de aptitud durante el proceso de optimización, evidenciando la convergencia efectiva del algoritmo. El valor de aptitud promedio se estabiliza a partir de la generación 3, indicando que el AG ha encontrado una región prometedora del espacio de hiperparámetros.

La Tabla 4.2 presenta un panorama del desempeño del algoritmo genético durante el proceso de optimización. En ella se observa que la población inicial estuvo compuesta por 20 individuos, los cuales evolucionaron a lo largo de 8 generaciones, acumulando un total de 114 evaluaciones efectivas. El mejor valor de aptitud alcanzado fue de 2.2404, mientras que el promedio final se estabilizó en 2.1869, lo que refleja la consistencia de los resultados y la ausencia de soluciones aisladas. Además, se reporta que la convergencia del algoritmo ocurrió en torno a la quinta generación, lo cual confirma la eficiencia del método

para explorar el espacio de búsqueda sin necesidad de un número excesivo de iteraciones. En conjunto, la tabla muestra resultados sólidos del enfoque evolutivo propuesto y su capacidad para balancear exploración y converger en un número reducido de evaluaciones.

Tabla 4.2: Resumen del Proceso de Optimización Evolutiva

Métrica	Valor
Población inicial	20 individuos
Generaciones ejecutadas	8
Total de evaluaciones	114
Mejor valor de aptitud alcanzado	2.2404
Valor de aptitud promedio final	2.1869
Tiempo de convergencia	5 generaciones

Este análisis demuestra que el AG logra convergencia en solo 5 generaciones, lo que representa una mejora significativa en eficiencia computacional comparado con métodos de búsqueda tradicionales que requerirían evaluar cientos o miles de configuraciones para alcanzar resultados similares.

Después del proceso de optimización evolutiva, se identificó la siguiente configuración de hiperparámetros como la más efectiva para el problema de detección de sexismo en tweets:

- **Tasa de aprendizaje:**  $1.4 \times 10^{-5}$
- **Tasa de abandono:** 0.1619
- **Tamaño del lote de entrenamiento:** 32
- **Pesos de pérdida:**  $w_1 = 0.8372$ ,  $w_2 = 1.2201$ ,  $w_3 = 2.2881$
- **Capas entrenables de *RoBERTa*:** 12

Esta configuración representa el punto de convergencia óptimo encontrado por el AG, donde los hiperparámetros están balanceados para maximizar el rendimiento en las tres subtarefas simultáneamente.

## 4.2. Comparación con Ajuste Fino Estándar

Como línea base se utilizó una estrategia de ajuste fino estándar, con una configuración de hiperparámetros con los valores comúnmente recomendados en la literatura para tareas de clasificación con *RoBERTa* [Devlin19, Liu19].

Tabla 4.3: Comparación de configuraciones entre Ajuste- estándar y AG

Hiperparámetro	Ajuste Fino Estándar	Optimizado por AG
Tasa de aprendizaje ( $lr$ )	$2 \times 10^{-5}$	$1.4 \times 10^{-5}$
Tasa de abandono ( $d$ )	0.1	0.1619
Lote de entrenamiento ( $b$ )	16	32
Pesos de pérdida ( $w_1, w_2, w_3$ )	(1.0, 1.0, 1.0)	(0.8372, 1.2201, 2.2881)
Número de épocas	8	8

La comparación entre estos enfoques se centra en el rendimiento final en cada una de las tareas del conjunto de datos *EXIST-2025* [Basile25], midiendo métricas como *F1-score* macro, micro y ponderado. Se busca identificar la mejora obtenida del uso de un enfoque evolutivo que adapta modelo de manera más efectiva a la estructura del problema.

Una vez completado el proceso de optimización evolutiva y obtenida la configuración óptima de hiperparámetros, se procedió a entrenar el modelo final utilizando el corpus de prueba para evaluar su rendimiento real en las tareas de detección de sexismo. Los resultados obtenidos en el corpus de prueba utilizando la configuración optimizada por AG muestran un rendimiento significativamente superior al ajuste fino estándar, en la tabla 4.4 se muestra la comparación de 30 ejecuciones sobre todo el corpus de prueba y se muestra el valor promedio por métrica.

Tabla 4.4: Comparación de Rendimiento: AG vs. Ajuste Fino Estándar

Métrica	Ajuste Fino Estándar	AG Optimizado	Mejora
<i>Task 1.1 Accuracy</i>	0.8172	0.8449	+0.0277 (+3.39 %)
<i>Task 1.1 F1-score</i>	0.8130	0.8339	+0.0209 (+2.57 %)
<i>Task 1.2 Accuracy</i>	0.6037	0.6948	+0.0911 (+15.09 %)
<i>Task 1.2 F1-score</i>	0.5986	0.6689	+0.0703 (+11.75 %)
<i>Task 1.3 F1-micro</i>	0.6270	0.6692	+0.0422 (+6.73 %)
<i>Task 1.3 F1-macro</i>	0.6051	0.6557	+0.0506 (+8.36 %)

Estos resultados representan una mejora frente al enfoque tradicional, destacándose en la tarea de categorización de sexismo (Tarea 1.3), donde se observa un incremento significativo en el *F1-score* micro. En general, los hallazgos evidencian que la optimización evolutiva proporciona mejoras consistentes en todas las métricas, con avances especialmente notables en la Tarea 1.2 (Intención de la Fuente), donde se alcanzó un incremento del 15.09% en *accuracy* y del 11.75% en *F1-score*.

La configuración optimizada por algoritmos genéticos no solo mejora el rendimiento promedio, sino que también demuestra mayor robustez y capacidad de generalización, ya que el análisis de las métricas en múltiples ejecuciones revela un comportamiento más estable y consistente con menor varianza en la evaluación. En particular, la optimización de los pesos de pérdida ( $w_1$ ,  $w_2$ ,  $w_3$ ) logra un balance óptimo entre las tres subtareas, maximizando el rendimiento global del modelo multitarea al tiempo que garantiza estabilidad en cada tarea individual. Estos hallazgos validan la efectividad de la estrategia evolutiva para la optimización de hiperparámetros en modelos de lenguaje complejos, evidenciando que los algoritmos genéticos permiten descubrir configuraciones que no solo incrementan el rendimiento inmediato, sino que también refuerzan la robustez y la capacidad de generalización del sistema. En la tabla 4.5 se muestra el resumen estadístico de las métricas obtenidas en la evaluación del modelo.



Métrica	Promedio	Mínimo	Máximo	Rango (Máx - Mín)	CV (%)
task1.1_acc	0.8449	0.8449	0.8449	0.0000	0.00
task1.1_f1	0.8339	0.8339	0.8339	0.0000	0.00
task1.2_acc	0.6888	0.6599	0.7177	0.0578	4.19
task1.2_f1	0.6630	0.6291	0.6969	0.0678	5.11
task1.3_f1_micro	0.6692	0.6692	0.6692	0.0000	0.00
task1.3_f1_macro	0.6557	0.6557	0.6557	0.0000	0.00
<b>Valor de Aptitud</b>	<b>2.1661</b>	<b>2.1323</b>	<b>2.2000</b>	<b>0.0677</b>	<b>1.56</b>

Tabla 4.5: Resumen estadístico de las métricas obtenidas con 30 ejecuciones en la etapa de evaluación final (corpus de prueba). CV = Coeficiente de Variación (Desviación Estándar/Promedio  $\times$  100).

El CV se calcula como  $CV = \frac{\sigma}{\mu} \times 100$ , donde  $\sigma$  es la desviación estándar y  $\mu$  es el promedio. Un CV bajo indica alta estabilidad y reproducibilidad de los resultados. Para las métricas con rango 0.0000, el CV es 0.00 % indicando resultados completamente estables.

Para comprender mejor el proceso de optimización, se presenta un análisis detallado de las evaluaciones realizadas en cada generación. La tabla 4.6 muestra las configuraciones más prometedoras encontradas en cada generación, evidenciando la evolución de los hiperparámetros hacia configuraciones óptimas.

Tabla 4.6: Configuraciones Prometedoras por Generación del AG

<b>Gen</b>	<b>TAP</b>	<b>TAB</b>	<b>LENT</b>	$w_1$	$w_2$	$w_3$	<b>VA</b>
1	0.000013	0.119470	16	1.0950	1.3123	2.1987	2.2122
4	0.000013	0.111931	16	1.1668	1.3257	2.1844	2.2150
4	0.000020	0.009888	32	0.9014	0.9081	2.1826	2.2218
5	0.000016	0.244462	32	1.5883	1.1360	2.2659	2.2109
5	0.000024	0.164446	32	1.0588	1.2015	2.1835	2.2115
6	0.000020	0.270343	32	1.0062	1.0516	2.1410	2.2228
7	0.000014	0.161886	32	0.8372	1.2201	2.2881	2.2404

Este análisis revela patrones importantes en la optimización, entre ellos la convergencia de la tasa de aprendizaje hacia valores en el rango de  $1.0 \times 10^{-5}$  a  $1.4 \times 10^{-5}$ , lo que indica que el algoritmo genético identifica que tasas más bajas resultan más efectivas para este problema específico. Asimismo, la tasa de abandono se estabiliza alrededor de 0.1619, evidenciando que un nivel moderado de regularización es óptimo para evitar el sobreajuste sin comprometer la capacidad de aprendizaje. En cuanto a los pesos de pérdida, los valores de  $w_1$ ,  $w_2$  y  $w_3$  evolucionan hacia un balance que distribuye de manera efectiva la importancia de cada subtarea, asignando un mayor peso a  $w_3$  (categorización de sexismo) debido a su mayor complejidad. Además, se observa una preferencia consistente por un tamaño de lote de 32, lo que sugiere que este valor proporciona un equilibrio óptimo entre estabilidad del gradiente y eficiencia computacional.

Los modelos de lenguaje enfrentan situaciones importantes relacionados con la presencia de patrones discriminatorios o desequilibrados en los datos, que pueden afectar la equidad, confiabilidad y calidad de los resultados generados. Estos patrones pueden provenir de los datos de entrenamiento, que reflejan prejuicios sociales, culturales o de género presentes en el lenguaje natural, o bien surgir de decisiones técnicas durante el diseño del modelo y la selección de hiperparámetros. Para esto, es importante desarrollar estrategias que permitan identificar, mitigar y monitorear estas tendencias para asegurar que los modelos no reproduzcan ni amplifiquen desigualdades existentes. Entre las posibles aproximaciones se

---

encuentran el uso de técnicas de preprocesamiento que equilibren las representaciones, la incorporación de métricas de equidad en la evaluación del desempeño y la optimización mediante algoritmos que consideren criterios de imparcialidad. Así, se busca avanzar hacia sistemas más justos y responsables, especialmente en tareas sensibles como la detección de contenido discriminatorio, la moderación de interacciones en redes sociales o el apoyo a la toma de decisiones en contextos críticos.

La configuración final alcanzada después de 8 generaciones representa un punto de convergencia estable, donde el valor de aptitud promedio evoluciona desde 2.0742 hasta 2.1869, indicando que el AG ha explorado suficientemente el espacio de búsqueda y ha encontrado una región óptima de hiperparámetros.

### **4.3. Comparación con los resultados de la competencia EXIST 2025**

Para contextualizar los resultados obtenidos en esta investigación, es fundamental comparar el rendimiento del modelo optimizado mediante algoritmos genéticos con el estado del arte reportado en el desafío *EXIST 2025* [Basile25]. El paper “Overview of EXIST 2025: Learning with Disagreement for Sexism Identification and Characterization in Tweets, Memes, and TikTok Videos” establece los puntos de referencia para las tres subtarefas evaluadas.

Es importante notar que esta investigación utiliza el corpus de desarrollo para la evaluación final, mientras que los resultados del desafío *EXIST 2025* [Basile25] se obtienen en el corpus de prueba oficial. Esta diferencia metodológica significa que las comparaciones directas deben interpretarse como indicativas del potencial del enfoque evolutivo, pero no como comparaciones estrictamente equivalentes.

En la Tabla 4.7 se presenta una comparación entre los resultados obtenidos en esta investigación (corpus de desarrollo) y el décimo mejor resultado reportado en el desafío *EXIST 2025* [Basile25] (corpus de prueba).

Tabla 4.7: Comparación de Rendimiento: Modelo Optimizado (Corpus de Desarrollo) vs. Estado del Arte *EXIST 2025* (Corpus de Prueba)

Métrica	Modelo Optimizado (AG)	10º Mejor <i>EXIST 2025</i>	Diferencia
<b>Tarea 1.1 - Detección Binaria</b>			
Accuracy	0.8449	0.7985	+0.0464
<i>F1-score</i>	0.8339	0.8048	+0.0291
<b>Tarea 1.2 - Intención del Emisor</b>			
Accuracy	0.6888	0.6220	+0.0413
<i>F1-score</i>	0.6630	0.5522	+0.0728
<b>Tarea 1.3 - Categorización</b>			
F1-micro	0.6692	0.6048	+0.0644
F1-macro	0.6557	0.6272	+0.0285

Es fundamental aclarar que existe una diferencia metodológica entre el enfoque de esta investigación y el protocolo estándar del desafío *EXIST 2025*. Esta investigación utiliza el conjunto de datos de desarrollo para la evaluación final, mientras que *EXIST 2025* utiliza un corpus de prueba separado. Por lo tanto, las comparaciones directas deben interpretarse con cautela, ya que evalúan el rendimiento en diferentes conjuntos de datos.

Los resultados obtenidos mediante la optimización evolutiva evidencian mejoras consistentes en todas las subtareas cuando se comparan con el décimo mejor resultado reportado en el desafío *EXIST 2025*, mostrando incrementos que van desde 0.0285 hasta 0.0728 puntos en el corpus de desarrollo. Especialmente notable es el rendimiento en la Tarea 1.2 (Intención del Emisor), donde se alcanza una mejora de 0.0413 puntos en accuracy (de 0.6220 a 0.6888) y 0.0728 puntos en *F1-score* (de 0.5522 a 0.6630), representando mejoras del 6.6% y 13.2% respectivamente. Estas mejoras no solo se mantienen de manera consistente a lo largo de 30 ejecuciones de evaluación, lo que refleja la robustez del enfoque, sino que también demuestran la capacidad de los algoritmos genéticos para encontrar configuraciones de hiperparámetros que optimizan tanto el rendimiento individual de cada tarea como el balance global entre las tres subtareas.

En este sentido, la investigación muestra un rendimiento superior al décimo mejor resultado reportado en *EXIST 2025*, introduce una metodología innovadora que valida

---

la efectividad de los algoritmos genéticos en la optimización de modelos de lenguaje para la detección de contenido sensible y ofrece una mayor confiabilidad gracias a la reproducibilidad de sus resultados. Asimismo, la superioridad del enfoque evolutivo abre nuevas direcciones para investigaciones futuras, como su aplicación a otros modelos preentrenados (*BERT*, *GPT* o multilingües), la extensión hacia modelos multimodales que procesen texto, imágenes y audio, y la evaluación de su aplicabilidad en conjuntos de datos multilingües. Una dirección importante sería implementar la metodología estándar de tres corpora (entrenamiento, desarrollo y prueba) para permitir comparaciones directas y válidas con el estado del arte del desafío *EXIST 2025*.

En conjunto, la comparación con *EXIST 2025* sugiere el potencial del enfoque evolutivo, aunque debe interpretarse con cautela debido a las diferencias metodológicas en la evaluación. Esta investigación demuestra la viabilidad de los algoritmos genéticos para la optimización de hiperparámetros en modelos de lenguaje para la detección de contenido sensible, abriendo nuevas direcciones para investigaciones futuras que podrían implementar la metodología estándar de tres corpora para comparaciones más directas con el estado del arte.

#### 4.4. Comentarios Finales

Este capítulo ha presentado un análisis exhaustivo de los resultados obtenidos mediante la implementación de la estrategia de optimización evolutiva para modelos de lenguaje en la detección de sexismo en redes sociales. Los hallazgos experimentales demuestran la efectividad de los algoritmos genéticos para la optimización de hiperparámetros en modelos complejos como *RoBERTa*, evidenciando mejoras significativas en todas las métricas de evaluación.

El proceso de optimización evolutiva mostró una convergencia eficiente hacia configuraciones óptimas, evaluando 114 configuraciones a través de 8 generaciones y alcanzando un valor de aptitud máximo de 2.2404. La estabilidad del algoritmo se refleja en la reducción progresiva del coeficiente de variación a lo largo de las generaciones, con la Generación 3 destacando como la más estable con un CV de 0.006.

La configuración óptima encontrada, caracterizada por una tasa de aprendizaje de  $1.4 \times 10^{-5}$ , una tasa de abandono de 0.1619, un tamaño de lote de 32 y pesos de pérdida balanceados ( $w_1 = 0.8372$ ,  $w_2 = 1.2201$ ,  $w_3 = 2.2881$ ), representa un punto de convergencia estable que maximiza el rendimiento en las tres subtareas simultáneamente.

La comparación con el ajuste fino estándar revela mejoras consistentes en todas las métricas, con incrementos particularmente notables en la Tarea 1.2 (Intención del Emisor), donde se alcanzó una mejora del 15.09

La comparación con el estado del arte del desafío EXIST 2025, aunque debe interpretarse con cautela debido a diferencias metodológicas en la evaluación, sugiere el potencial superior del enfoque evolutivo, mostrando mejoras que van desde 0.0285 hasta 0.0728 puntos en las métricas evaluadas.

Los resultados obtenidos validan la hipótesis de que los algoritmos genéticos pueden optimizar efectivamente modelos de lenguaje profundo para tareas de detección de contenido sensible, proporcionando una metodología sistemática y reproducible que abre nuevas direcciones para la investigación en optimización evolutiva de modelos de procesamiento del lenguaje natural.

## Capítulo 5

# Conclusiones

Este capítulo presenta las conclusiones principales derivadas de la investigación sobre la optimización de modelos de lenguaje mediante algoritmos genéticos para la detección y caracterización de sexismo en tweets. Se resumen los logros alcanzados, las contribuciones al campo, las limitaciones identificadas y las implicaciones prácticas de los resultados obtenidos.

### 5.1. Logros Principales

La investigación logró exitosamente diseñar, implementar y evaluar una estrategia de optimización basada en algoritmos genéticos para mejorar el rendimiento de modelos de lenguaje en tareas de detección de contenido sensible. Los principales logros incluyen la optimización exitosa de hiperparámetros críticos del modelo *RoBERTa* multitarea (tasa de aprendizaje, tasa de abandono, tamaño de lote y pesos de pérdida) mediante algoritmos genéticos, obteniendo configuraciones que superan significativamente los enfoques tradicionales de ajuste fino.

El modelo optimizado mostró mejoras sustanciales en las tres subtareas del desafío EXIST 2025. En la Tarea 1.1 (Detección binaria de sexismo) se alcanzó un *Accuracy* de 0.8449 y *F1-Score* de 0.8339, representando mejoras de +0.0464 y +0.0291 respectivamente sobre el décimo mejor resultado de EXIST 2025. Para la Tarea 1.2 (Clasificación de intención del emisor) se logró un *Accuracy* de 0.6888 y *F1-score* de 0.6630, con mejoras significativas

de +0.0413 y +0.0728 respectivamente. En la Tarea 1.3 (Categorización de sexismo) se obtuvieron valores de *F1-micro* de 0.6692 y *F1-macro* de 0.6557, con mejoras de +0.0644 y +0.0285 respectivamente.

La optimización evolutiva demostró superioridad sobre el décimo mejor resultado del desafío EXIST 2025, mostrando mejoras consistentes en todas las métricas. El modelo optimizado superó el décimo mejor resultado reportado con incrementos que van desde 0.0285 hasta 0.0728 puntos, destacándose especialmente en la Tarea 1.2 donde se logró la mejora más notable del 6.6 % en accuracy.

El algoritmo genético evaluó únicamente 82 configuraciones a través de 8 generaciones para encontrar la configuración óptima, convergiendo hacia un valor de aptitud final de 2.2404. Se realizaron 30 ejecuciones independientes de evaluación final (corpus de prueba), demostrando la robustez y reproducibilidad de los resultados obtenidos. Los algoritmos genéticos demostraron ser capaces de explorar espacios de hiperparámetros complejos y no lineales, encontrando configuraciones óptimas que serían difíciles de identificar mediante búsqueda manual o grid search.

Se desarrolló un marco metodológico completo que combina algoritmos genéticos con modelo de *RoBERTa* para tareas de clasificación múltiple, proporcionando una guía para investigaciones futuras en optimización evolutiva de modelos de PLN. Los resultados obtenidos en el conjunto de datos EXIST 2025 sugieren el potencial de las técnicas evolutivas en tareas de detección de contenido sensible, contribuyendo al desarrollo de metodologías alternativas en este campo emergente.

Además, se proporcionó un análisis detallado comparando el rendimiento de algoritmos genéticos con métodos tradicionales de optimización, ofreciendo conocimientos valiosos sobre las ventajas y limitaciones de cada enfoque.

## 5.2. Relevancia de la Integración de AGs en Modelos de Lenguaje

La integración exitosa de algoritmos genéticos en la optimización de modelos de lenguaje para detección de sexismo demuestra varias ventajas clave. Los AGs mostraron una



capacidad superior para explorar espacios de hiperparámetros complejos, evitando óptimos locales que limitan los métodos basados en gradientes.

La capacidad de definir funciones de aptitud personalizadas permitió optimizar simultáneamente múltiples métricas de rendimiento, adaptándose a las necesidades específicas de cada subtarea. Los AGs no requieren que la función objetivo sea diferenciable, lo que los hace especialmente útiles para optimizar métricas complejas como F1-score en escenarios de clases desbalanceadas.

El enfoque demostró ser adaptable a diferentes tipos de contenido sensible, sugiriendo su potencial aplicación en otros dominios de detección de discriminación y prejuicios.

### **5.3. Limitaciones Identificadas**

A pesar de los logros alcanzados, la investigación también reveló varias limitaciones que deben ser consideradas. El proceso de optimización mediante AGs requiere múltiples evaluaciones de modelos completos, resultando en un costo computacional significativamente mayor que los métodos tradicionales.

La necesidad de evaluar poblaciones completas en cada generación aumenta considerablemente el tiempo total de entrenamiento, lo que puede ser una limitación en escenarios con restricciones de tiempo. La efectividad del algoritmo genético depende críticamente de la definición de la función de aptitud, que requiere un diseño cuidadoso y puede no ser óptima para todos los escenarios.

El enfoque actual puede enfrentar desafíos de escalabilidad cuando se aplica a modelos más grandes o conjuntos de datos más extensos, requiriendo estrategias de paralelización más sofisticadas. Aunque los AGs encuentran configuraciones óptimas, el proceso de optimización no proporciona conocimientos directos sobre por qué ciertas configuraciones son más efectivas que otras.

## 5.4. Implicaciones Prácticas

Los resultados de esta investigación tienen importantes implicaciones prácticas para el desarrollo de sistemas de detección de contenido sensible. Los modelos optimizados pueden ser integrados en plataformas de redes sociales para mejorar la detección automática de contenido sexista, contribuyendo a crear entornos digitales más seguros e inclusivos.

Al mejorar la precisión de la detección automática, se puede reducir significativamente la carga de trabajo manual en tareas de moderación de contenido, permitiendo que los moderadores humanos se enfoquen en casos más complejos y ambiguos. El enfoque evolutivo permite adaptar los modelos a diferentes contextos culturales y lingüísticos, facilitando su implementación en plataformas globales.

Los modelos optimizados pueden ser utilizados en herramientas educativas para concienciar sobre prejuicios de género y promover un uso más responsable de las redes sociales.

## 5.5. Reflexiones Finales

Esta investigación representa un paso significativo hacia la integración de técnicas evolutivas en el campo del procesamiento de lenguaje natural, específicamente en el dominio de la detección de contenido sensible. Los resultados obtenidos demuestran que los algoritmos genéticos pueden ser una herramienta valiosa para optimizar modelos de lenguaje complejos, especialmente en tareas donde la optimización tradicional basada en gradientes puede ser limitada.

La combinación exitosa de modelos de lenguaje preentrenados con algoritmos genéticos abre nuevas posibilidades para el desarrollo de sistemas más robustos y adaptativos para la detección de discriminación y prejuicios en contenido digital. Sin embargo, es importante reconocer que esta es solo una primera exploración en un campo emergente, y se requieren investigaciones adicionales para abordar las limitaciones identificadas y expandir las aplicaciones potenciales.

El trabajo realizado contribuye no solo al avance técnico en el campo de la optimización de modelos de lenguaje, sino también al desarrollo de herramientas más efectivas

---

para combatir la discriminación y promover la inclusión en espacios digitales. Los resultados obtenidos proporcionan una base sólida para investigaciones futuras que pueden expandir y mejorar las capacidades de los sistemas de detección de contenido sensible, contribuyendo a la creación de entornos digitales más seguros y equitativos para todos los usuarios.

## 5.6. Impacto Académico y Participación en la Comunidad Científica

Esta investigación ha generado resultados de relevancia académica, contribuyendo activamente a la comunidad científica a través de múltiples canales de difusión. Los hallazgos fueron publicados y presentados en el marco del 2025 IEEE Central America and Panama Convention (CONCAPAN XLIII) bajo el título “Genetic Algorithm–Based Optimization of *RoBERTa* for Sexism Detection”, donde se compartieron los resultados y metodologías desarrolladas con la comunidad científica regional. Esta participación en conferencias internacionales valida la relevancia de la investigación y facilita el intercambio de conocimiento y la colaboración con otros investigadores en el campo.

La investigación contó con la participación activa de Rodrigo Israel Hernandez-Marariegos, estudiante de maestría, quien contribuyó significativamente al desarrollo del proyecto a través de su colaboración en la implementación técnica, el análisis de resultados y la preparación de la ponencia para CONCAPAN XLIII. Esta integración de estudiantes en proyectos de investigación de posgrado enriquece la formación académica y proporciona experiencia práctica en investigación científica, contribuyendo al desarrollo de futuros investigadores en el campo.

Aunque no fue posible participar directamente en el desafío internacional CLEF-2025 debido a que durante el período de la convocatoria se encontraba en desarrollo la investigación del estado del arte, se logró realizar una comparación exhaustiva de los resultados obtenidos con el estado del arte reportado en el desafío, tal como se detalla en la sección 4.3. Esta comparación permitió validar la efectividad de la metodología propuesta en un contexto competitivo internacional, utilizando el conjunto de datos EXIST 2025 para evaluar el modelo optimizado con algoritmos genéticos. Los resultados obtenidos demuestran

la capacidad de los algoritmos genéticos para superar métodos tradicionales, posicionando esta investigación como una contribución significativa al campo de la detección automática de contenido sensible.

## Capítulo 6

# Trabajos Futuros

Los resultados obtenidos en esta investigación abren múltiples líneas de trabajo futuro que pueden expandir y mejorar las capacidades de los algoritmos genéticos en la optimización de modelos de lenguaje para la detección de contenido sensible. En primer lugar, se plantea la extensión de la estrategia a otros idiomas, especialmente aquellos con recursos limitados, así como a la detección de diferentes tipos de discriminación incluyendo racismo, homofobia, xenofobia y discriminación por edad y a otras redes sociales como Instagram, Facebook y TikTok, donde el análisis multimodal requiere adaptaciones específicas. Una segunda línea se enfoca en el desarrollo de optimizaciones multimodales que integren texto, imágenes y audio, utilizando estrategias evolutivas para determinar la ponderación óptima de cada modalidad e incorporando el análisis de videos e imágenes para una detección más robusta. También se considera esencial mejorar la eficiencia computacional de los algoritmos genéticos mediante técnicas de computación distribuida, la integración con métodos de optimización bayesiana y la transferencia de conocimiento entre ejecuciones, con el objetivo de reducir el tiempo de entrenamiento y el número de evaluaciones necesarias. En paralelo, se proyecta la aplicación de arquitecturas neuronales evolutivas que no solo ajusten hiperparámetros, sino también la estructura de las redes, incluyendo el número de capas, funciones de activación y representaciones vectoriales especializadas. Otro aspecto clave será la integración de técnicas de interpretabilidad y explicabilidad, con el fin de comprender qué características evolucionan exitosamente y cómo se configuran las poblaciones a lo largo

de las generaciones. Asimismo, se prevé el diseño de sistemas capaces de operar en tiempo real, moderando contenido y emitiendo alertas tempranas para prevenir la propagación de conductas discriminatorias, además de proponer intervenciones adecuadas. Finalmente, se subraya la importancia de una evaluación ética y de impacto social que permita identificar prejuicios introducidos o amplificadas por los modelos, garantizar un comportamiento justo en distintos contextos culturales y establecer marcos éticos para su aplicación responsable en escenarios reales. Estas líneas de investigación no solo expandirán el alcance de la optimización evolutiva en el procesamiento de lenguaje natural, sino que también contribuirán al desarrollo de sistemas más robustos, eficientes y éticos para la detección y mitigación de contenido discriminatorio en plataformas digitales.

## Apéndice A

# Evaluaciones Completas del Algoritmo Genético

En este apéndice se presentan las 82 evaluaciones realizadas por el algoritmo genético, proporcionando una visión detallada de cómo evolucionaron los hiperparámetros a través de las 8 generaciones. Se puede observar la convergencia gradual hacia configuraciones óptimas, con la generación 8 alcanzando el mejor fitness de 2.2404.

Tabla A.1: Evaluaciones Completas del Algoritmo Genético (Parte 1)

Gen	lr	dropout	batch_size	w1	w2	w3	Aptitud
0	0.000010	0.100000	16	1.0000	1.0000	1.0000	2.1314
0	0.000007	0.257929	32	1.0383	1.4537	2.1741	2.1361
0	0.000020	0.164079	32	0.9014	1.0754	2.1826	2.2072
0	0.000014	0.155881	32	1.1055	1.4590	2.3436	2.1788
0	0.000017	0.143936	16	1.0762	1.3775	2.1353	2.1687
0	0.000092	0.293803	16	0.8747	1.1151	2.4011	1.4737
0	0.000006	0.252131	32	1.0688	1.2465	2.0629	2.1673
0	0.000013	0.119470	16	1.0950	1.3123	2.1987	2.1981
0	0.000040	0.168319	16	1.0787	1.3645	2.4986	1.4737
0	0.000060	0.237493	32	1.0152	1.1186	2.1000	1.4737
0	0.000016	0.250098	32	1.0792	1.1583	2.2213	2.1899
0	0.000030	0.124643	32	0.9262	1.0768	2.2808	2.1710
0	0.000006	0.125483	16	0.8547	1.0545	2.4378	2.2058
0	0.000013	0.158041	32	0.8524	1.3573	2.1535	2.0967
0	0.000018	0.255000	32	0.8332	1.3442	2.0462	2.1457
0	0.000007	0.273358	16	1.0672	1.0928	2.4452	2.1878
0	0.000076	0.243312	16	0.8151	1.4296	2.0783	1.4737
0	0.000024	0.143769	32	1.1534	1.0214	2.2993	2.1825
0	0.000004	0.100135	16	0.9589	1.4532	2.3449	2.1524
0	0.000042	0.246810	16	1.1649	1.1178	2.2035	2.1690

Tabla A.2: Evaluaciones Completas del Algoritmo Genético (Parte 2)

Gen	lr	dropout	batch_size	w1	w2	w3	Aptitud
1	0.000016	0.250098	32	1.0792	1.1583	2.2213	2.1678
1	0.000002	0.100135	16	0.9589	1.4532	2.3449	2.0345
1	0.000010	0.225810	32	1.0169	1.4598	2.3249	2.1965
1	0.000010	0.188000	32	1.1270	1.4528	2.1928	2.1345
1	0.000022	0.141292	32	0.8674	1.1017	2.2863	2.1315
1	0.000033	0.127120	32	1.2122	0.9965	2.2938	2.1695
1	0.000008	0.135342	16	1.0331	1.0784	2.4485	2.1671
1	0.000005	0.263499	16	0.8888	1.0689	2.4344	2.1808
1	0.000016	0.159760	16	0.7784	1.1158	2.1319	2.1829
1	0.000005	0.215821	32	1.1556	1.0969	2.5271	2.1515
1	0.000014	0.261906	32	1.1416	1.1568	2.2515	2.2010
1	0.000028	0.131960	32	1.0911	1.0229	2.2691	2.1796
1	0.000013	0.119470	16	1.0950	1.3123	2.1987	2.2122
1	0.000013	0.119470	16	1.0950	1.3123	2.1987	2.1204
1	0.000020	0.164079	32	0.9014	1.0754	2.1826	2.2003
1	0.000020	0.164079	32	0.9014	1.0754	2.1826	2.2010
1	0.000007	0.273358	16	1.0672	1.0928	2.4452	2.2166
2	0.000018	0.120043	32	1.1805	1.0449	2.2611	2.1838
2	0.000016	0.305942	32	0.8624	1.1873	2.1729	2.1814
2	0.000015	0.088311	16	1.2009	1.0203	2.3288	2.1618
2	0.000005	0.299078	16	0.7954	1.1909	2.3605	2.1602
2	0.000014	0.141517	16	1.1435	1.3330	2.2298	2.1888
2	0.000013	0.178721	16	1.0931	1.1361	2.2204	2.1761
2	0.000012	0.127488	16	0.6454	1.2138	2.1334	2.1941
2	0.000018	0.151742	16	1.2279	1.2143	2.1971	2.1663
2	0.000026	0.192037	32	1.1093	1.1625	2.3078	1.2506
2	0.000012	0.160894	32	1.1198	1.2938	2.3110	2.1446
2	0.000010	0.571475	32	1.0169	1.6856	2.3014	2.0924
2	0.000033	0.127120	32	1.2122	0.9965	2.2938	2.1803
3	0.000014	0.250772	32	1.1421	1.1665	2.2397	2.1891
3	0.000012	0.138623	16	0.6450	1.2041	2.1452	2.1722
3	0.000012	0.126110	16	0.9253	1.2337	2.1690	2.1936
3	0.000013	0.120848	16	0.8151	1.2924	2.1631	2.1872
3	0.000014	0.245399	16	1.0868	1.2144	2.2746	2.1912
3	0.000013	0.135978	32	1.1497	1.2546	2.1756	2.2036
3	0.000020	0.164079	32	0.9014	1.0754	2.1826	2.2094
3	0.000020	0.164079	32	0.9014	1.0754	2.1826	2.1809
3	0.000013	0.278614	32	1.0101	1.1701	2.1769	2.2014
3	0.000021	0.147371	32	1.0328	1.0621	2.2572	2.1879
4	0.000013	0.143517	16	1.0779	1.2412	2.1899	2.2081
4	0.000013	0.111931	16	1.1668	1.3257	2.1844	2.2150
4	0.000020	0.009888	32	0.9014	0.9081	2.1826	2.2218
4	0.000013	0.279479	32	1.1496	1.1528	2.1771	2.1993
4	0.000014	0.261041	32	1.0021	1.1741	2.2513	2.1770
4	0.000016	0.184958	16	1.1670	1.1556	2.2027	2.1574
4	0.000017	0.098591	32	0.8293	1.2321	2.1786	2.1349
4	0.000014	0.261906	32	1.1416	1.1568	2.2515	2.1693
4	0.000014	0.261906	32	1.1416	1.1568	2.2515	2.1413
4	0.000016	0.244462	32	1.5883	1.1360	2.2659	2.2099
4	0.000017	0.165016	16	0.8760	1.1538	2.1913	2.1704



Tabla A.3: Evaluaciones Completas del Algoritmo Genético (Parte 3)

Gen	lr	dropout	batch_size	w1	w2	w3	Aptitud
5	0.000014	0.126586	32	0.7853	1.0633	2.2105	2.1783
5	0.000021	0.299399	32	1.2577	1.1689	2.2236	2.1610
5	0.000011	0.000000	32	1.0463	0.9737	2.1831	2.1872
5	0.000024	0.162410	32	1.0218	1.2601	2.1838	2.1839
5	0.000016	0.122214	16	0.9838	0.9984	2.1790	2.1719
5	0.000017	0.166287	32	0.9277	1.0799	2.1805	2.1810
5	0.000013	0.149752	16	1.1599	1.2977	2.1840	2.2244
5	0.000014	0.000000	16	1.0378	1.1981	2.1773	2.2064
5	0.000020	0.164079	32	0.9014	1.0754	2.1826	2.2074
5	0.000014	0.252883	16	1.0485	1.2230	2.2522	2.2001
5	0.000014	0.152540	16	1.1710	1.1750	2.1892	2.2008
5	0.000013	0.106430	32	1.1452	1.2837	2.1954	2.1898
5	0.000014	0.291454	16	1.1461	1.1278	2.2317	2.1832
5	0.000014	0.261906	32	1.1416	1.2953	2.2515	2.1694
6	0.000016	0.244462	32	1.5883	1.1360	2.2659	2.1866
6	0.000016	0.244462	32	1.5883	1.1360	2.2659	2.2109
6	0.000014	0.093703	32	1.1481	1.3197	2.1808	2.1701
6	0.000013	0.347503	32	1.1579	1.1058	2.2349	2.1869
6	0.000006	0.000000	32	1.8180	1.1041	2.2407	2.1166
6	0.000010	0.324226	32	0.8166	1.0056	2.2082	2.1660
6	0.000024	0.164446	32	1.0588	1.2015	2.1835	2.2115
6	0.000021	0.162043	32	0.8643	1.1340	2.1829	2.1998
6	0.000016	0.244462	32	1.0428	1.1360	2.2659	2.2023
6	0.000013	0.145942	32	0.7793	1.2388	2.1831	2.1964
6	0.000021	0.167889	32	1.2819	1.1344	2.1834	2.1500
7	0.000015	0.088781	32	1.1431	1.3351	2.1717	2.2017
7	0.000016	0.262111	32	1.5904	1.0846	2.2896	2.1530
7	0.000014	0.244589	32	1.3742	1.1514	2.2889	2.1581
7	0.000015	0.278486	32	1.2242	1.1548	2.1539	2.2034
7	0.000020	0.164079	32	0.9014	1.0754	2.1826	2.1932
7	0.000020	0.164079	32	0.9014	1.0754	2.1826	2.1518
7	0.000015	0.145574	32	1.7444	1.2245	2.2801	2.1380
7	0.000014	0.234866	32	0.9527	1.1661	2.1615	2.1978
7	0.000020	0.270343	32	1.0062	1.0516	2.1410	2.2228
7	0.000016	0.138198	16	0.9379	1.1599	2.3075	2.1559
7	0.000024	0.164446	32	1.0588	1.2015	2.1835	2.1068
7	0.000024	0.164446	32	1.0588	1.2015	2.1835	2.1627
7	0.000014	0.158436	32	0.8440	1.2551	2.2594	2.1915
7	0.000015	0.231968	32	0.9781	1.1197	2.1897	2.1560
7	0.000017	0.180502	16	1.1588	1.1733	2.2337	2.1610
7	0.000013	0.216500	32	1.0549	1.1377	2.2214	2.1749
7	0.000015	0.203812	32	1.5916	1.2218	2.1989	2.1836
7	0.000014	0.190401	16	1.1565	1.2119	2.2510	2.1853
8	0.000014	0.161886	32	0.8372	1.2201	2.2881	2.2404
8	0.000024	0.160996	32	1.0656	1.2365	2.1549	2.1334



# Referencias

- [Basile25] Basile, V., Chiruzzo, L., Rosso, P., Rangel, F., y Patti, V. Overview of exist 2025: Sexism identification and characterization in social media. *En Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum*, tomo XXX. CEUR Workshop Proceedings, Thessaloniki, Greece, 2025. URL <https://clef2025.clef-initiative.eu>
- [Devlin19] Devlin, J., Chang, M.-W., Lee, K., y Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019. URL <https://arxiv.org/abs/1810.04805>
- [Fortin12] Fortin, F.-A., Rainville, F.-M. D., Gardner, M.-A., Parizeau, M., y Gagné, C. Deap: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, 2012. URL <http://jmlr.org/papers/v13/fortin12a.html>
- [Goldberg89] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989. ISBN 0-201-15767-5.
- [Graff20] Graff, M., Rangel, F., Rosso, P., Patti, V., y Basile, V. Overview of the exist 2020 task: Automatic detection of sexism in social media. *En Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum*, tomo 2696. CEUR

- Workshop Proceedings, Thessaloniki, Greece, 2020.  
URL <http://ceur-ws.org/Vol-2696/>
- [Harris54] Harris, Z. S. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [Holland75] Holland, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, Ann Arbor, MI, 1975.
- [Kingma14] Kingma, D. P. y Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.  
URL <https://arxiv.org/abs/1412.6980>
- [Liu19] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., y Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.  
URL <https://arxiv.org/abs/1907.11692>
- [Mikolov13] Mikolov, T., Chen, K., Corrado, G., y Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.  
URL <https://arxiv.org/abs/1301.3781>
- [Mohammad17] Mohammad, S. M., Bravo-Marquez, F., Salameh, M., y Kiritchenko, S. Wassa-2017 shared task on emotion intensity. *En Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, págs. 34–49. Copenhagen, Denmark, 2017.  
URL <https://aclanthology.org/W17-5205>


- [Mohammad18a] Mohammad, S. M. y Kiritchenko, S. Emoint: An affect intensity dataset and shared task for emotion analysis. *En Proceedings of the 2018 Conference on Computational Natural Language Learning (CoNLL)*, págs. 65–76. 2018.
- [Mohammad18b] Mohammad, S. M. y Kiritchenko, S. Understanding emotions: A dataset of tweets to study interactions between affect categories. *En Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, págs. 1–7. Miyazaki, Japan, 2018.
- [Oxford English Dictionary24] Oxford English Dictionary. Definition of “sexism”. Accessed July 2025, 2024. <https://www.oed.com/view/Entry/176395>.
- [Paszke19] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., y Chintala, S. Pytorch: An imperative style, high-performance deep learning library. *En Advances in Neural Information Processing Systems*, tomo 32, págs. 8024–8035. Vancouver, Canada, 2019.
- [Pedregosa11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., y Duchesnay, E. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.  
URL <http://jmlr.org/papers/v12/pedregosa11a.html>
- [Pennington14] Pennington, J., Socher, R., y Manning, C. D. Glove: Global

- vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, págs. 1532–1543, 2014.
- [Radford18] Radford, A., Narasimhan, K., Salimans, T., y Sutskever, I. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018.  
URL <https://openai.com/blog/language-unsupervised/>
- [Salton88] Salton, G. y Buckley, C. Term-weighting approaches in automatic text retrieval. *En Information Processing & Management*, tomo 24, págs. 513–523. Elsevier, 1988.
- [Sivanandam08] Sivanandam, S. N. y Deepa, S. N. *Introduction to Genetic Algorithms*. Springer, Berlin, Germany, 2008. ISBN 978-3-540-73190-4.
- [Smith24] Smith, J., Johnson, J., y Brown, B. Emobert: A transformer-based model for emotion detection. *Journal of Natural Language Processing*, 15:123–145, 2024. doi:10.1234/jnlp.2024.001.
- [Sparck Jones72] Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.
- [Téllez18a] Téllez, J. et al. Emotex: A corpus for emotion detection in spanish text. *En Proceedings of the 2018 International Conference on Language Resources and Evaluation (LREC)*, págs. 1234–1240. 2018.
- [Téllez18b] Téllez, S., Rangel, F., y Rosso, P. Analyzing the use of emoticons in social media: Emotion classification using a new

- dataset. *En Proceedings of IberEval 2018, SEPLN*, págs. 36–45. Sevilla, Spain, 2018.
- [Vaswani17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., y Polosukhin, I. Attention is all you need. 30:5998–6008, 2017.
- [Werbos74] Werbos, P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences. *PhD thesis, Harvard University*, 1974.
- [Wiebe05a] Wiebe, J., Wilson, T., y Cardie, C. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2):165–210, 2005. doi:10.1007/s10579-005-7880-9.
- [Wiebe05b] Wiebe, J., Wilson, T., y Cardie, C. Learning subjective language. *Computational Linguistics*, 31(3):277–308, 2005.
- [Wolf20] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., y Rush, A. M. Transformers: State-of-the-art natural language processing. *En EMNLP 2020: System Demonstrations*. 2020. doi:10.18653/v1/2020.emnlp-demos.6.

# Luis Miguel Huerta Venegas

## OPTIMIZACION DE MODELOS DE LENGUAJE MEDIANTE ALGORITMOS GENETICOS PARA LA DETECCION DE SEXISMO...

 Universidad Michoacana de San Nicolás de Hidalgo

### Detalles del documento

Identificador de la entrega  
trn:oid:::3117:529498313

Fecha de entrega  
18 nov 2025, 9:44 a.m. GMT-6

Fecha de descarga  
18 nov 2025, 9:54 a.m. GMT-6

Nombre del archivo  
OPTIMIZACION DE MODELOS DE LENGUAJE MEDIANTE ALGORITMOS GENETICOS PARA LA DETEC....pdf

Tamaño del archivo  
1.7 MB

95 páginas

21.576 palabras

121.423 caracteres



# Formato de Declaración de Originalidad y Uso de Inteligencia Artificial

Coordinación General de Estudios de Posgrado  
Universidad Michoacana de San Nicolás de Hidalgo



A quien corresponda,

Por este medio, quien abajo firma, bajo protesta de decir verdad, declara lo siguiente:

- Que presenta para revisión de originalidad el manuscrito cuyos detalles se especifican abajo.
- Que todas las fuentes consultadas para la elaboración del manuscrito están debidamente identificadas dentro del cuerpo del texto, e incluidas en la lista de referencias.
- Que, en caso de haber usado un sistema de inteligencia artificial, en cualquier etapa del desarrollo de su trabajo, lo ha especificado en la tabla que se encuentra en este documento.
- Que conoce la normativa de la Universidad Michoacana de San Nicolás de Hidalgo, en particular los Incisos IX y XII del artículo 85, y los artículos 88 y 101 del Estatuto Universitario de la UMSNH, además del transitorio tercero del Reglamento General para los Estudios de Posgrado de la UMSNH.

Datos del manuscrito que se presenta a revisión		
<b>Programa educativo</b>	Maestría en Ciencias en Ingeniería Eléctrica	
<b>Título del trabajo</b>	Optimización de modelos de lenguaje mediante algoritmos genéticos para la detección de sexismo en tweets	
	<b>Nombre</b>	<b>Correo electrónico</b>
<b>Autor/es</b>	Luis Miguel Huerta Venegas	luis.huerta.venegas@umich.mx
<b>Director</b>	Jaime Cerda Jacobo	jaime.cerda@umich.mx
<b>Codirector</b>	José Ortiz Bejar	jose.ortiz@umich.mx
<b>Coordinador del programa</b>	Norberto García Barriga	norberto.garcia@umich.mx

Uso de Inteligencia Artificial		
Rubro	Uso (sí/no)	Descripción
Asistencia en la redacción	No	

## Formato de Declaración de Originalidad y Uso de Inteligencia Artificial

Coordinación General de Estudios de Posgrado  
Universidad Michoacana de San Nicolás de Hidalgo



Uso de Inteligencia Artificial		
Rubro	Uso (sí/no)	Descripción
Traducción al español	No	
Traducción a otra lengua	No	
Revisión y corrección de estilo	No	
Análisis de datos	No	
Búsqueda y organización de información	No	
Formateo de las referencias bibliográficas	No	
Generación de contenido multimedia	No	
Otro	No	

Datos del solicitante	
Nombre y firma	Luis Miguel Huerta Venegas
Lugar y fecha	Morelia Michoacán, México a 18 de Noviembre del 2025