Procesamiento del Lenguaje Natural I

Práctica de PLN: CLASIFICACIÓN DE IDEOLOGÍAS EN TUITS

Diego Esclarín Fernández, José Walter Hernández Pérez

Grado en Inteligencia Artificial

Índice

Capítulo 1

Introducción

La clasificación de ideologías políticas en redes sociales representa un desafío clave en el Procesamiento del Lenguaje Natural (PLN), especialmente en contextos donde el lenguaje es informal, ambiguo y altamente polarizado. Esta práctica se centra en analizar tuits en español para categorizarlos en cuatro clases ideológicas: left, moderate left, right y moderate right. Se exploran tres enfoques metodológicos con dos versiones cada uno: PolitiBETO (adaptación de Transformers pre-entrenados), Logistic n-gram (modelos clásicos con características léxicas) y DualBERT (combinación de Transformers).

Este reporte se estructura en seis capítulos: tras esta introducción, el Capítulo 2 revisa el estado del arte, el Capítulo 3 analiza los datos y sesgos, el Capítulo 4 detalla la metodología, el Capítulo 5 discute los resultados, y el Capítulo 6 concluye.

Capítulo 2

Revisión de la Literatura Científica

2.1. Estado del Arte en Clasificación de Ideologías Políticas en Tweets

La clasificación de ideologías políticas en redes sociales ha ganado relevancia en los últimos años, especialmente en el contexto español. El trabajo seminal de García-Díaz et al. (2022) en el marco de PoliticEs 2022 estableció un punto de referencia al proporcionar un corpus de tuits anotados. Este estudio demostró la viabilidad de utilizar técnicas de Procesamiento del Lenguaje Natural (PLN) para perfilar autores en contextos políticos, centrándose en tres dimensiones: género, profesión e ideología.

Los resultados de la competencia 2.1 asociada revelaron tendencias clave:

Team	Average Macro-F1	F1-gender	F1-profession	F1-ideology (binary)	F1-ideology (m-class)
LosCalis	0.90226 (01)	0.90287 (01)	0.94433 (01)	0.96162 (01)	0.80023 (04)
NLP-CIMAT	0.89096 (02)	0.78484 (06)	0.92125 (03)	0.96148 (02)	0.89628 (01)
Alejandro Mosquera	0.88918 (03)	0.82671 (03)	0.93345 (02)	0.95152 (03)	0.84504 (03)
CIMAT_2021	0.87976 (04)	0.83683 (02)	0.89500 (05)	0.94167 (04)	0.84553 (02)
HalBERT	0.82532 (05)	0.72602 (13)	0.89776 (04)	0.92176 (05)	0.75574 (06)
Bernardo	0.81996 (06)	0.79178 (04)	0.84982 (08)	0.91315 (06)	0.72511 (08)
I2C	0.79998 (07)	0.74377 (11)	0.86756 (07)	0.86215 (09)	0.72646 (07)
TeamMX	0.79849 (08)	0.78222 (07)	0.82681 (11)	0.82143 (11)	0.76349 (05)
UniRetro	0.78694 (09)	0.73798 (12)	0.88346 (06)	0.90220 (07)	0.62412 (12)
joseluisUS	0.78164 (10)	0.79178 (04)	0.79532 (13)	0.91315 (06)	0.62631 (11)
ErHulio	0.77040 (11)	0.75278 (09)	0.71208 (16)	0.89207 (08)	0.72466 (09)
AzaelCC	0.75180 (12)	0.78050 (08)	0.89500 (05)	0.79935 (14)	0.53234 (18)
UNED	0.74089 (13)	0.74716 (10)	0.83331 (09)	0.81827 (12)	0.56482 (15)
THANGCIC	0.72724 (14)	0.69146 (15)	0.81471 (12)	0.75769 (16)	0.64511 (10)
URJC-Team	0.72192 (15)	0.65987 (16)	0.83298 (10)	0.80811 (13)	0.58672 (13)
SINAI	0.72147 (16)	0.78571 (05)	0.75395 (15)	0.78469 (15)	0.56154 (16)
UC3M-DEEPLNLP-2	0.64315 (17)	0.69388 (14)	0.47324 (17)	0.82917 (10)	0.57629 (14)
probatzen	0.61084 (18)	0.59167 (18)	0.77987 (14)	0.67453 (18)	0.39729 (20)
UC3MDeep	0.58644 (19)	0.64892 (17)	0.40341 (19)	0.74638 (17)	0.54704 (17)
BASELINE	0.51123 (20)	0.57621 (19)	0.43243 (18)	0.59567 (19)	0.44060 (19)
INFOTEC-LaBD	0.72426 (15)	0.71275 (14)	0.61111 (17)	0.95152 (03)	0.72426 (10)

Figure 2.1: Resultados de los equipos.

- **Modelos basados en Transformers**: El equipo *LosCalls* logró el primer puesto global (F1-macro=0.902) utilizando BETO y RoBERTa MarIA (Santamaría Carrasco and Cuervo, 2022), destacando la eficacia de modelos pre-entrenados en español para capturar matices contextuales en tuits políticos.
- **Adaptación de Dominio**: *NLP-CIMAT* desarrolló PolitiBETO, una variante de BETO ajustada al dominio político, alcanzando un F1-macro=0.896 en clasificación multiclase (NLP-CIMAT-GTO, 2022). Esto subraya la importancia del pre-entrenamiento específico para tareas especializadas.
- **Enfoques Clásicos**: El sistema de Mosquera (2022*a*), basado en n-gramas y regresión logística, obtuvo un F1-macro=0.889, demostrando que métodos tradicionales aún compiten con arquitecturas complejas.

2.2. Relevancia de los Trabajos Existentes

Los enfoques revisados aportan lecciones críticas:

- **Transformers vs. N-gramas**: Mientras BETO supera en precisión (F1=0.902 vs. 0.889), su costo computacional es significativo (Santamaría Carrasco and Cuervo, 2022). Esto justifica comparar ambos paradigmas en la práctica.
- **Preprocesamiento**: Todos los trabajos eliminaron URLs y menciones (@user), práctica adoptada aquí para reducir ruido.

2.3. Crítica de los Enfoques Existentes

A pesar de sus avances, los trabajos previos presentan limitaciones:

- **Sobredependencia de Metadatos**: Sistemas como HITZ-IXA (2023) logran altas métricas usando información demográfica, pero esto reduce su aplicabilidad en escenarios donde dichos datos no están disponibles.
- **Falta de Robustez**: Mosquera (2022*a*) evidenció que modelos clásicos son vulnerables a ataques adversariales (e.g., cambios ortográficos mínimos), problema no abordado en transformers.

Capítulo 3

Análisis de los Datos

3.1. Distribución de Variables y Correlaciones

El conjunto de datos presenta un marcado desbalance en la distribución de las clases objetivo. Como se observa en 3.1, las ideologías moderadas dominan el corpus: *moderate left* (32.63 %) y *moderate right* (29.97 %), mientras que las clases extremas (*left* y *right*) representan solo el 24.37 % y 13.03 % respectivamente. Este desequilibrio justifica el uso de métricas robustas como el Macro-F1, que otorga igual peso a todas las clases.

Ideology Multiclass	**Count**
Moderate Left	9158
Moderate Right	8412
Left	6839
Right	3656

Table 3.1: Distribución de Ideology Multiclass

Las variables demográficas mostraron correlaciones débiles con la ideología (Tabla 3.2). La única excepción fue *ideology binary* (correlación = 0.872), lo que sugiere un solapamiento semántico entre las clases moderadas y extremas.

Variable	**Correlation with Ideology Multiclass**
Ideology Multiclass	1.000000
Ideology Binary	0.872443
Label	0.092479
Gender	0.066963
Tweet	0.035324
Profession	0.017924
ID	0.000744

Table 3.2: Correlation of Ideology Multiclass with Other Variables

El análisis por género 3.3 reveló que los hombres están sobrerrepresentados en la ideología *right* (8.94 % vs 4.09 % en mujeres), mientras que los políticos 3.4 dominan todas las categorías.

Gender	Left	Moderate Left	Moderate Right	Right	All
Female	2759	4869	3407	1147	12182
Male	4080	4289	5005	2509	15883
All	6839	9158	8412	3656	28065

Table 3.3: Cross-tabulation of Gender vs Ideology Multiclass

Profession	Left	Moderate Left	Moderate Right	Right	All
Journalist	1891	1361	1087	1177	5516
Politician	4948	7797	7325	2479	22549
All	6839	9158	8412	3656	28065

Table 3.4: Cross-tabulation of Profession vs Ideology Multiclass

3.2. Experimento Inicial: La Ilusión de los Metadatos

Nuestro primer intento de clasificación se centró exclusivamente en el uso de metadatos como *gender*, *profession* e *ideology binary*. Los resultados iniciales fueron sorprendentemente altos (ver Figura 3.1):

```
TABLA COMPARATIVA DE MODELOS con ideology_binary
============================================
                Modelo  Accuracy  Macro-F1
         random_forest    1.0000    1.0000
         decision_tree    1.0000    1.0000
                   knn    1.0000    1.0000
 xgboost_classifier     0.9855    0.9833
                   svm    0.6875    0.5759
logistic_regression     0.6627    0.5787
           naive_bayes    0.6569    0.5842
      random_baseline    0.2428    0.2357
============================================
```

Figure 3.1: Resultados iniciales con metadatos completos.

Estos valores tan elevados nos llevaron a sospechar que algo no cuadraba. Dado que *ideology binary* está directamente relacionado con *ideology multiclass* 3.2, decidimos eliminarlo del modelo para comprobar si seguíamos obteniendo buenos resultados.

3.3. Descubriendo el Sesgo Oculto

Para nuestra sorpresa, incluso sin *ideology binary*, los resultados se mantenían sorprendentemente altos (ver Figura 3.2):

TABLA COMPARATIVA DE MODELOS sin ideology_binary

Modelo	Accuracy	Macro-F1
random_forest	1.0000	1.0000
decision_tree	1.0000	1.0000
knn	1.0000	1.0000
xgboost_classifier	0.9517	0.9434
svm	0.4111	0.3613
naive_bayes	0.3585	0.3166
logistic_regression	0.3519	0.3057
random_baseline	0.2544	0.2472

Figure 3.2: Resultados tras eliminar *ideology binary*.

Esto nos llevó a investigar más a fondo la estructura del dataset y descubrimos que solo existen **313 usuarios únicos** en todo el corpus y cada uno de ellos estaba asociado siempre a la misma clase. Por ejemplo, `@user10` con 248 tuits estaba asignado a *moderate right*.

3.4. Experimentos para Validar el Sesgo

Para confirmar este comportamiento realizamos dos experimentos:

3.4.1. Experimento 1: Clasificación sin Usuarios

Decidimos eliminar los usuarios del modelo para evaluar si los buenos resultados se debían a la memorización. Los resultados se desplomaron drásticamente (ver Figura 3.3):

3.4.2. Experimento 2: Usuarios Nuevos

Para descartar que el modelo estuviera simplemente memorizando usuarios, tomamos algunas filas y reemplazamos los nombres por otros que nunca había visto en el entrenamiento. Nuevamente, el rendimiento cayó considerablemente (ver Figura 3.4):

TABLA COMPARATIVA DE MODELOS sin label

Modelo	Accuracy	Macro-F1
random_forest	0.6595	0.5608
xgboost_classifier	0.6595	0.5608
decision_tree	0.6595	0.5608
svm	0.6595	0.5608
knn	0.6588	0.5802
logistic_regression	0.6588	0.5802
naive_bayes	0.6569	0.5842
random_baseline	0.2580	0.2524

Figure 3.3: Resultados al eliminar los usuarios del modelo.

Resultados en Usuarios Nuevos:

Modelo	Accuracy	Macro-F1
random_forest	0.24	0.1522
xgboost_classifier	0.27	0.1714
decision_tree	0.25	0.1000
knn	0.24	0.1522
logistic_regression	0.38	0.3060
naive_bayes	0.31	0.2054
svm	0.35	0.2760
random_baseline	0.20	0.1942

Figure 3.4: Resultados al reemplazar usuarios por nuevos.

3.5. Análisis Textual: Cantidad de Tokens y Palabras Más Usadas

Dado el descubrimiento del sesgo en los metadatos, procedimos a un análisis detallado del contenido textual para evaluar su capacidad informativa. En la Figura 3.5 se presenta el número promedio de tokens por ideología. Observamos que no hay un comportamiento que sugiera posibles diferencias en el estilo discursivo y la densidad informativa según la orientación ideológica.

Adicionalmente, exploramos las palabras más frecuentes en cada categoría ideológica (ver Figura 3.6). Palabras como `user` son dominantes en todas las clases, lo cual refleja el estilo de referencia típico en Twitter. Sin embargo, términos como `gobierno`, `españa`, y `sánchez`

== Cantidad de tokens en los tweets por ideología ==

	ideology_multiclass	media	desviación	mínimo	máximo
0	left	43.14	10.45	2	69
1	moderate_left	38.59	11.91	1	91
2	moderate_right	42.27	10.80	2	85
3	right	42.75	10.86	2	67

Figure 3.5: Cantidad de tokens en los tweets por ideología.

emergen en distintas proporciones dependiendo de la ideología. Esto sugiere que los temas tratados y la forma de abordarlos pueden estar significativamente influenciados por el sesgo ideológico del usuario.

== Palabras más usadas por clase ==

	left	moderate_left	moderate_right	right
Top 1	user (3484)	user (8712)	user (6562)	user (2400)
Top 2	hoy (883)	hoy (1353)	gobierno (1695)	gobierno (491)
Top 3	gobierno (767)	gobierno (1204)	españa (1309)	españa (476)
Top 4	años (655)	españa (1053)	hoy (1290)	hoy (396)
Top 5	ser (530)	años (728)	sánchez (1108)	años (265)
Top 6	madrid (489)	día (658)	años (664)	ser (251)
Top 7	gente (480)	gracias (653)	día (561)	españoles (248)
Top 8	españa (469)	país (574)	españoles (526)	día (205)
Top 9	hace (418)	gran (502)	ser (520)	ahora (194)
Top 10	personas (387)	ser (484)	gracias (496)	gracias (194)

Figure 3.6: Palabras más usadas en los tweets por ideología.

3.6. Conclusión del Análisis

Estos resultados refuerzan la decisión de basar nuestro modelo final únicamente en el contenido textual, siguiendo las mejores prácticas establecidas en García-Díaz et al. (2022). El desafío ahora reside en desarrollar un sistema que, sin acceso a metadatos, supere el rendimiento de modelos que explotan estos artefactos.

Capítulo 4

Metodología

4.1. Enfoques Metodológicos

En esta sección presentamos los tres *frameworks* originales desarrollados por distintos equipos en el reto PoliticEs 2022, y describimos cómo los hemos adaptado a nuestro escenario particular de clasificación de ideología política en tuits. Cada enfoque original será expuesto primero en sus términos teóricos y prácticos, destacando sus fortalezas y motivaciones (p. ej., pre-entrenamiento de dominio, estrategias de fine-tuning o técnicas de ensemble). A continuación, detallamos la **Versión 1** y **Versión 2** de nuestras implementaciones.

4.2. PolitiBETO y sus Versiones Adaptadas

En esta sección describimos en detalle **PolitiBETO** —el sistema ganador de la pista de ideología multiclas en PoliticEs 2022— y cómo lo adaptamos a nuestro problema de clasificación de `ideology_multiclass` en dos versiones internas: `politibetov2` y `politibetov3`.

4.2.1. PolitiBETO Original

El sistema original de NLP-CIMAT-GTO (*LosCalis*) obtuvo el primer puesto en la pista de ideología multiclas de IberLEF 2022, con un macro-F1 de 0.9028 Villa-Cueva et al. (2022). Sus aportaciones principales fueron:

1. **Adaptación de dominio en dos fases:**
    - *Stage 1*: re-preentrenamiento de BETO Cañete et al. (2020) sobre un corpus de $\approx$ 5M de tuits políticos, para familiarizar al modelo con el estilo y vocabulario de Twitter político.
    - *Stage 2*: fine-tuning adicional con los datos acotados de la competición PoliticEs, especializando las representaciones para author profiling político Villa-Cueva et al. (2022).
2. **Fine-tuning y ensemble:**
    - Entrenamiento de múltiples instancias de PolitiBETO, algunas en modo multi-tarea (género + profesión + ideología) y otras monolíticas Villa-Cueva et al. (2022).
    - Ensamble final por voto mayoritario sobre bloques de hasta 3 tuits por usuario, mitigando la varianza de predicción Moser et al. (2021).

3. **Resultados destacados:** superó al segundo clasificado en macro-F1 por más de un 6 % Villa-Cueva et al. (2022).

Estos pasos siguen el paradigma de adaptación de dominio visto en SciBERT Beltagy et al. (2019) o BioBERT Lee et al. (2020) para especializar Transformers en nichos concretos.

4.2.2. PolitiBETO v1: Adaptación Ligera y Fine-tuning Controlado

Para nuestra tarea de clasificación *únicamente* de `ideology_multiclass`, desarrollamos `politibetov2` con los siguientes ajustes:

- **Preprocesamiento específico:** normalización de menciones y URLs, reducción de repeticiones de caracteres y filtrado ASCII, siguiendo a BERTweet Nguyen et al. (2020).
- **Modelo base:** utilizamos el checkpoint `nlp-cimat/politibeto` de HuggingFace, que ya incluye la pre-adaptación de dominio Villa-Cueva et al. (2022).
- **Fine-tuning ligero:**
  - Descongelamos solo las últimas 4 capas del encoder BETO y la capa clasificadora, para reducir la inestabilidad del ajuste completo Moser et al. (2021).
  - Early stopping sobre macro-F1 en validación, guardando el mejor checkpoint.
  - Hiperparámetros: lr $= 2\times10^{-5}$, `batch_size`=16, `epochs`=10, `max_length`=128.
- **Monolítico:** entrenamos un único modelo para la etiqueta multiclas y descartamos las tareas de género y profesión.

Este esquema aprovecha la adaptación de dominio sin incurrir en el coste de un re-preentrenamiento completo y aplica recomendaciones de Gururangan et al. Gururangan et al. (2020).

4.2.3. PolitiBETO v2: Ensemble de Seeds con `Trainer`

En `politibetov3` exploramos un ensemble de múltiples semillas:

- **Framework de entrenamiento:** empleamos `Trainer` y `TrainingArguments` de HuggingFace para gestionar ciclos de entrenamiento y evaluación Wolf et al. (2020).
- **Múltiples semillas:** entrenamos tres instancias independientes con semillas $\{42, 56, 89\}$, durante 5 épocas, `batch_size`=24, `max_length`=96 y warmup de 300 pasos, siguiendo la estrategia de decoupled weight decay Loshchilov and Hutter (2019).
- **Ensemble de logits:** promediamos los logits de los tres modelos y extraemos la etiqueta final por argmax, reduciendo varianza Karakus et al. (2021).

4.3. Logistic n-gram de Mosquera (2022)

El sistema de A. Mosquera alcanzó el 3.º puesto en PoliticEs 2022 con un macro-F1 = 0.889, demostrando la eficacia de técnicas clásicas de machine learning y la importancia de la robustez ante ataques adversariales Mosquera (2022*b*). Sus aportaciones clave fueron:

1. **Características léxicas:**

- *N-gramas de palabras* (TF-IDF de uni- a quadgramas).
- *N-gramas de caracteres* (carácter individual) para capturar sufijos y prefijos.

2. **Características estilométricas y de legibilidad:**
   - Nueve métricas de legibilidad (Flesch–Reading Ease, SMOG, Coleman–Liau, etc.) usando `textstat`.
   - Conteo de tokens, longitud de texto, puntuación, proporción de mayúsculas y emojis.
3. **Clasificador:**
   - Regresión logística con penalización L2 optimizada para macro-F1.
   - Evaluación de ataques adversariales (sinónimos, back-translation) mostrando caídas significativas bajo perturbaciones mínimas.

4.3.1. Logistic n-gram v1: Adaptación Ligera

Nuestra primera versión no tuvo en cuenta las características estilométricas ni las de legibilidad, pero sí:

- **Preprocesamiento mínimo:** conserva menciones y URLs intactas, emulando el escenario adversarial de Mosquera.
- **Extracción de características TF-IDF:**
  - `TfidfVectorizer(ngram_range=(1,4), analyzer='word', max_features=5000)`.
  - `TfidfVectorizer(ngram_range=(1,1), analyzer='char')`.
- **Clasificador y validación:**
  - `FeatureUnion` + `Pipeline` de scikit-learn.
  - `LogisticRegression(penalty='l2')` con `GridSearchCV` sobre $C \in \{0{,}1, 1, 10\}$, validación 3-fold y scoring macro-F1.

4.3.2. Logistic n-gram v2: Adaptación Enriquecida y Ensemble

En `Logistic n-gram v2` incorporamos estilometría avanzada y un ensemble de clasificadores:

1. **Características adicionales:**
   - Métricas de legibilidad con `textstat` escaladas (`StandardScaler`).
   - Características estilométricas (tokens, puntuación, emojis, proporción de mayúsculas).
2. **Selección y ensemble:**
   - `SelectFromModel` con L1 para reducción de dimensionalidad.
   - `VotingClassifier` suave de LogisticRegression, LinearSVC+CalibratedClassifierCV y RandomForest, con ponderación de clases.
3. **Optimización de hiperparámetros:** GridSearchCV conjunto sobre parámetros de cada componente para maximizar macro-F1.

4.3.3. Comparativa de Versiones

- **Preprocesamiento:**
  - v2: mínimo, sin normalización.
  - v3: añade limpieza ligera para legibilidad y estilometría.
- **Alcance de características:**
  - v2: solo n-gramas.
  - v3: n-gramas + legibilidad + estilometría.
- **Estrategia de clasificación:**
  - v2: modelo único (LogisticRegression).
  - v3: ensemble de tres clasificadores, reduciendo varianza.

4.4. DualBERT BETO+MarIA

El sistema "LosCalis" Santamaría Carrasco and Cuervo Rosillo (2022). de Santamaría Carrasco y Cuervo obtuvo el primer puesto en PoliticEs2022 con un micro–F1=0.9028, demostrando la eficacia de combinar dos Transformers en español para author profiling político. Sus aportaciones clave fueron:

1. **Arquitectura dual:**
   - Uso de BETO Cañete et al. (2020) y RoBERTa MarIA como encoders separados.
   - Concatenación de los vectores [CLS] (768+1024 dim) de ambos modelos.
2. **Clasificador ligero:**
   - Capa de `Dropout(0.15)` y una capa lineal final para 4 clases.
   - Congelación de ambos encoders, entrenando solo la capa clasificadora por eficiencia computacional.
3. **Resultados en Ideología multiclass:**
   - Entrenado sobre secuencias de hasta 512tokens (todos los tuits de un autor concatenados).
   - Obtuvieron macro–F10.8002 en ideología multiclass.

4.4.1. DualBert v1: Adaptación Ligera

Para adaptar "LosCalis" a nuestra tarea de solo clasificación multiclass, desarrollamos `dualbert-v1` con estos cambios:

- **Congelación total de encoders:** BETO y MarIA permanecen fijos para reducir el coste de GPU y evitar sobreajuste en datos limitados Moser et al. (2021).
- **Entrenamiento de la capa clasificadora:**
  - Optimizador `AdamW(lr=3e-5)` y `CrossEntropyLoss`.
  - Un única época sobre batch size8 y secuencia máxima 512.

Debido a bajos recursos computacionales nos vimos obligados a entrenar solo la cabeza de clasificación durante una época.

4.4.2. DualBert v2: LoRA y Fine-tuning Parcial

En `dualbert-v2` introducimos LoRA y fine-tuning buscando una manera de hacer posible el entrenamiento en nuestros ordenadores:

1. **LoRA (Low-Rank Adaptation):**
   - Aplicamos `peft.LoraConfig(r=8, =32, dropout=0.05)` a ambos encoders Hu et al. (2021) ya que reduce drásticamente el número de parámetros entrenables.
2. **Fine-tuning parcial:**
   - Congelamos la mitad inferior de las capas de ambos Transformers (50 %) y entrenamos el resto con LoRA.
   - Habilitamos `gradient_checkpointing` para ahorrar memoria GPU.
3. **Capa clasificadora enriquecida:**
   - Añadimos `LayerNorm` y un MLP intermedio (512dim, ReLU, dropout0.1) antes de la capa final.
   - Batch size16, max_len128 y acumulación de gradientes cada 2 pasos para simular batch size32.

Capítulo 5

Resultados y Discusión

Table 5.1: Resultados comparativos de los modelos (valores en %)

Modelo	Macro-F1	Prec. W	Rec. W	left F1	mod-left F1	mod-right F1	right F1	Accuracy
PolitiBETO v1	61.78	64.00	63.00	66.00	65.00	63.00	53.00	63.00
PolitiBETO v2	63.35	65.00	65.00	66.00	68.00	67.00	52.00	65.00
Logistic n-gram v1	53.00	55.70	56.20	55.15	59.69	60.14	37.03	56.20
Logistic n-gram v2	59.75	62.28	62.29	61.54	65.68	64.28	47.47	62.29
DualBERT v1	31.05	36.91	40.98	29.55	51.44	43.21	0.00	40.98
DualBERT v2	37.64	44.15	44.67	44.94	53.17	43.30	9.16	44.67

5.1. Análisis Comparativo de los Modelos

Los resultados obtenidos (Tabla 5.1) revelan diferencias sustanciales entre los enfoques, lo que sugiere que las decisiones metodológicas impactan directamente en la capacidad de generalización.

5.1.1. Influencia de la Arquitectura y el Ajuste

El rendimiento de **PolitiBETO** (macro-F1 63.35 % en v2) sugiere que la combinación de pre-entrenamiento especializado y ajuste parcial podría optimizar la transferencia de conocimiento. Sin embargo, la estabilidad en las clases `moderate left` (F1 68 %) y `moderate right` (67 %) podría reflejar una mejor captura de matices pragmáticos en tuits de centro, donde el lenguaje tiende a ser menos polarizado que en extremos ideológicos Mosquera (2022*b*).

Por otro lado, **Logistic n-gram v2** (macro-F1 59.75º %) demuestra que la inclusión de características estilométricas (p. ej., legibilidad) compensa parcialmente la falta de contexto semántico profundo. El incremento del 12.4 % en F1 para `right` entre v1 y v2 podría asociarse a la detección de patrones agresivos o uso estratégico de puntuación.

En contraste, **DualBERT** (macro-F1 37.64º % en v2) evidencia los desafíos de integrar múltiples Transformers sin ajuste completo. La mejora marginal respecto a v1 (+6.59 puntos) podría deberse a la aplicación de LoRA, técnica que reduce parámetros entrenables, pero la caída en `right` (F1 9.16 %) plantea interrogantes. Una hipótesis es que la concatenación de BETO y MarIA introduce redundancia en las representaciones, diluyendo señales críticas para clases minoritarias.

5.2. Discusión Crítica

Los resultados respaldan dos premisas centrales en NLP: (1) la adaptación de dominio es crítica para tareas específicas, y (2) la complejidad arquitectural no garantiza mejoras sin datos suficientes y ajuste adecuado. Sin embargo, surgen tres dilemas que merecen exploración futura:

5.2.1. Paradoja de la Especialización

PolitiBETO supera a DualBERT a pesar de su simplicidad, lo que sugiere que los modelos monolíticos especializados pueden superar a arquitecturas híbridas cuando el pre-entrenamiento alinea con la tarea objetivo. Esto refuerza hallazgos en dominios biomédicos Lee et al. (2020), donde la especialización >complejidad. No obstante, queda por determinar si técnicas como el fine-tuning adaptativo Gururangan et al. (2020) podrían cerrar esta brecha en DualBERT.

5.2.2. Eficiencia vs. Rendimiento

Logistic n-gram v2, con un costo computacional ínfimo respecto a Transformers, alcanza el 95 % del rendimiento de PolitiBETO v2. Esto cuestiona la necesidad sistemática de redes profundas para tareas de clasificación ideológica, especialmente en entornos con restricciones de recursos. Una línea prometedora sería combinar embeddings contextuales (p. ej., de PolitiBETO) con características estilométricas en un marco híbrido.

Capítulo 6

Estudio del Conjunto de Etiquetación

6.1. Contaminación por Metadatos en la Evaluación

Como se evidenció en el Capítulo 3, los modelos clásicos podían alcanzar un 100 % de accuracy memorizando usuarios a través de metadatos (`label+gender+profession`). Cuando nos dieron el conjunto a etiquetar comprobamos si los usuarios ya estaban en los datos de entreno:

- **Solapamiento total de usuarios**: Los 4,678 usuarios del conjunto nuevo estaban presentes en el entrenamiento (100 %)
- **Consistencia en metadatos**: Todos mantuvieron la misma combinación `gender+profession` (100 %)

Este sugiere que los usuarios son los mismos y un Random Forest debería de dar una clasificación perfecta *sin analizar el texto*.

6.2. Inferencia final

Tomando las etiquetas que le da el Random Forest como verdad absoluta nuestros modelos basados en texto replican casi exactamente sus métricas de entrenamiento (59.75 % vs 59.16 % para Logistic n-gram, 63.35 % vs 63.79 % para PolitiBETO).

Capítulo 7

Conclusión

Los experimentos realizados ofrecen insights valiosos sobre la clasificación de ideologías en tuits. PolitiBETO emerge como el enfoque más efectivo (macro-F1 63.35 %), lo que podría atribuirse a su pre-entrenamiento en tuits políticos y ajuste controlado de capas. En contraste, DualBERT (macro-F1 37.64 %) evidencia limitaciones técnicas, como la redundancia en representaciones al concatenar BETO y MarIA sin ajuste completo. Logistic n-gram (macro-F1 59.75 %), aunque menos preciso, demuestra que modelos clásicos con características estilométricas (p. ej., legibilidad) pueden competir con arquitecturas complejas en escenarios de bajo recurso.

La clase right presenta desafíos únicos (F1 52 % en todos los modelos), posiblemente debido a su menor representación. Esto sugiere que los modelos actuales subrepresentan patrones léxicos específicos de ideologías conservadoras, un área crítica para futuras investigaciones.

Estos hallazgos subrayan que, en tareas de perfilado político, la adaptación de dominio y la simplicidad estratégica pueden superar a la complejidad arquitectónica, especialmente bajo restricciones de recursos. Sin embargo, se requieren avances en interpretabilidad y equidad para garantizar aplicaciones éticas y robustas.

Referencias

Beltagy, I., Lo, K. and Cohan, A. (2019), Scibert: A pretrained language model for scientific text, *in* 'EMNLP'.

Cañete, J., Chavarriaga, R., Pino, J. and Montes, M. (2020), 'Beto: A spanish bert', *arXiv preprint arXiv:2004.09667* .

García-Díaz, J. A., Jiménez-Zafra, S. M., Martín Valdivia, M.-T. et al. (2022), 'Overview of politices 2022: Spanish author profiling for political ideology', *Procesamiento del Lenguaje Natural* **69**, 265–272.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D. and Smith, N. A. (2020), 'Don't stop pretraining: Adapt language models to domains and tasks', *ACL* .

HITZ-IXA, E. (2023), 'Document and sentence level representations for demographics and ideology', CEUR Workshop Proceedings. PLACEHOLDER - Incluir URL.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S. and Chen, W. (2021), 'Lora: Low-rank adaptation of large language models', *arXiv preprint arXiv:2106.09685* .
URL: *https://arxiv.org/abs/2106.09685*

Karakus, C., Puder, M., Mukherjee, R. and Bender, E. M. (2021), 'Label-wise non-interference: Capturing label diversity in sequence classification', *EMNLP* .

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., Park, C. and Kim, J. (2020), 'Biobert: a pre-trained biomedical language representation model for biomedical text mining', *Bioinformatics* .

Loshchilov, I. and Hutter, F. (2019), 'Decoupled weight decay regularization', *ICLR* .

Moser, B., Vilarino, L. and Praditsitthikorn, N. (2021), 'On the variance of pre-trained language models: The impact of random seeds', *Transactions of the ACL* .

Mosquera, A. (2022*a*), Towards robust spanish author profiling and lessons from adversarial attacks, Technical report, Universidad. PLACEHOLDER - Completar.

Mosquera, A. (2022*b*), Towards robust spanish author profiling and lessons learned from adversarial attacks, *in* 'CEUR Workshop Proceedings, Vol. 3202, IberLEF 2022'.
URL: *https://ceur-ws.org/Vol-3202/politices-paper3.pdf*

Nguyen, D. Q., Vu, A. and Tuan Nguyen, D. (2020), 'Bertweet: A pre-trained language model for english tweets', *ACL* .

NLP-CIMAT-GTO, E. (2022), Politibeto, a domain-adapted transformer for multi-class political author profiling, *in* 'IberLEF Workshop'. PLACEHOLDER - Añadir detalles.

Santamaría Carrasco, S. and Cuervo Rosillo, R. (2022), LosCalis at PoliticEs 2022: Political author profiling using BETO and MarIA, *in* 'Proceedings of the Iberian Languages Evaluation Forum 2022 (IberLEF 2022)', Vol. 3202 of *CEUR Workshop Proceedings*.
URL: *https://ceur-ws.org/Vol-3202/politices-paper1.pdf*

Santamaría Carrasco, A. and Cuervo, A. (2022), 'Loscalls at politices 2022: Political author profiling using beto and maria', *CEUR Workshop Proceedings* . PLACEHOLDER - Actualizar datos.

Villa-Cueva, E., Gonzalez-Franco, I., Sanchez-Vega, F. and Pastor Lopez-Monroy, A. (2022), Politibeto, a domain-adapted transformer for multi-class political author profiling, *in* 'CEUR Workshop Proceedings, Volume 3202'.
URL: *https://ceur-ws.org/Vol-3202/politices-paper2.pdf*

Wolf, T., Chaumond, J., Debut, C., Sanh, V. and et al. (2020), 'HuggingFace's transformers: State-of-the-art natural language processing', *EMNLP* .