

GUILLERMO PALOMERO BERNAL

MEMORIA.pdf

-  Turnitin Memoria Final
 -  TFG ETSIINF (Moodle PP)
 -  Universidad Politecnica de Madrid
-

Document Details

Submission ID

trn:oid:::1:3458293426

93 Pages

Submission Date

Jan 15, 2026, 9:40 PM GMT+1

23,819 Words

Download Date

Jan 15, 2026, 9:45 PM GMT+1

139,333 Characters

File Name

11987_GUILLEMO_PALOMERO_BERNAL_MEMORIA_530462_93630407.pdf

File Size

5.5 MB

12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Top Sources

- 11%  Internet sources
 - 4%  Publications
 - 0%  Submitted works (Student Papers)
-

Top Sources

- 11% Internet sources
4% Publications
0% Submitted works (Student Papers)
-

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| Rank | Type | Source | Percentage |
|------|----------|------------------------------|------------|
| 1 | Internet | oa.upm.es | 3% |
| 2 | Internet | zaguan.unizar.es | <1% |
| 3 | Internet | uvadoc.uva.es | <1% |
| 4 | Internet | www.parallelparliament.co.uk | <1% |
| 5 | Internet | gredos.usal.es | <1% |
| 6 | Internet | www.mdpi.com | <1% |
| 7 | Internet | issuu.com | <1% |
| 8 | Internet | arxiv.org | <1% |
| 9 | Internet | www.researchgate.net | <1% |
| 10 | Internet | hdl.handle.net | <1% |
| 11 | Internet | sedici.unlp.edu.ar | <1% |

| | | |
|----|--|-----|
| 12 | Internet | |
| | www.coursehero.com | <1% |
| 13 | Internet | |
| | ru.dgb.unam.mx | <1% |
| 14 | Publication | |
| | Anabel Pilicita, Enrique Barra. "Using of Transformers Models for Text Classificati... | <1% |
| 15 | Internet | |
| | www.slideshare.net | <1% |
| 16 | Internet | |
| | www.informeraxen.es | <1% |
| 17 | Internet | |
| | www.dailymail.co.uk | <1% |
| 18 | Publication | |
| | McClung, James. "Simulations of Perpendicular Electron Holes.", University of Ne... | <1% |
| 19 | Publication | |
| | Morales Hernández, Roberto Carlos. "Reconocimiento de tendencias en un camp... | <1% |
| 20 | Internet | |
| | dspace.umh.es | <1% |
| 21 | Internet | |
| | www.springerprofessional.de | <1% |
| 22 | Publication | |
| | Abdullah Al Foysal, Ronald Böck. "Who Needs External References?—Text Summa... | <1% |
| 23 | Publication | |
| | Mª Empar Llorca Martínez. "Modificaciones químicas y microestructurales en la el... | <1% |
| 24 | Publication | |
| | Favreau, Charles-Olivier. "Learning to Rank with BERT for Argument Quality", Eco... | <1% |
| 25 | Internet | |
| | pure.rug.nl | <1% |

26 Publication

Alvarez, Carlos. "Aplicacion de Procesamiento de Lenguaje Natural Sobre Una Enc... <1%

27 Publication

Yuting Lin. "Legitimizing Negative Aspects in Corporate Social Responsibility Rep... <1%

28 Internet

ijsrcseit.com <1%

29 Internet

www.dykinson.com <1%

30 Internet

ceur-ws.org <1%

31 Internet

www.jove.com <1%

32 Publication

Elena Dominguez Romero, Jelena Bobkina. "Epistemic, emotional, and effective st... <1%

33 Publication

Gabriela Clemente Polo. "Efecto de la contracción en la cinética de secado de mús... <1%

34 Internet

digibuo.uniovi.es <1%

35 Internet

pingpdf.com <1%

36 Internet

www.un.org <1%

37 Internet

reliscoen.grupolys.org <1%

38 Publication

Phenikaa University <1%

39 Internet

aidipe2019.aidipe.org <1%

| | | |
|----|---|-----|
| 40 | Internet | |
| | www.ijsrcseit.com | <1% |
| 41 | Publication | |
| | Francisco Jurado. "Journalistic transparency using CRFs to identify the reporter of..." | <1% |
| 42 | Internet | |
| | repository.uel.ac.uk | <1% |
| 43 | Publication | |
| | "Applied Technologies", Springer Science and Business Media LLC, 2023 | <1% |
| 44 | Internet | |
| | internetpasoapaso.com | <1% |
| 45 | Internet | |
| | academica-e.unavarra.es | <1% |
| 46 | Internet | |
| | vdocumento.com | <1% |
| 47 | Internet | |
| | www.jourlib.org | <1% |
| 48 | Publication | |
| | Moy Rivera, Jorge Carlos. "Ruido de aeropuertos: Estudio del problema del ruido e..." | <1% |
| 49 | Internet | |
| | archive.org | <1% |
| 50 | Internet | |
| | bibliometria.ucm.es | <1% |
| 51 | Internet | |
| | etd.repository.ugm.ac.id | <1% |
| 52 | Internet | |
| | orbi.umons.ac.be | <1% |
| 53 | Internet | |
| | pubs.wi-kassel.de | <1% |

| | | |
|----|--|-----|
| 54 | Internet | |
| | redined.educacion.gob.es | <1% |
| 55 | Internet | |
| | www.encarte.org | <1% |
| 56 | Internet | |
| | www.monografias.com | <1% |
| 57 | Internet | |
| | www.univalle.edu.co | <1% |
| 58 | Publication | |
| | María Elena Placencia, Alejandro Parini. "Introducción al Estudio del Discurso Digi... | <1% |
| 59 | Publication | |
| | Rathore, Archit. "Topological Data Analysis and Visualization for Interpretable Ma... | <1% |
| 60 | Publication | |
| | Silvia Marzal Romeu. "Concepción e integración de arquitecturas y protocolos de ... | <1% |
| 61 | Publication | |
| | Víctor Javier Garrido Peñalver. "Arquitectura de interoperabilidad para mejorar la... | <1% |
| 62 | Publication | |
| | de Pinho, Carolina Andreia da Silva Martins. "Comunicações Sem Fios Para Sistem... | <1% |
| 63 | Internet | |
| | doaj.org | <1% |
| 64 | Internet | |
| | ruidera.uclm.es | <1% |
| 65 | Internet | |
| | www.cne.gov.do | <1% |
| 66 | Internet | |
| | www.fellerate.cl | <1% |
| 67 | Internet | |
| | www.um.edu.ar | <1% |

68 Publication

I. García-Díaz, F. Puertas, M. F. Gazulla, M. P. Gómez, M. Palacios. "Effect of ZnO; Z... <1%

69 Internet

code.cacert.org <1%

70 Internet

dspace.unl.edu.ec <1%

71 Internet

elartu.tntu.edu.ua <1%

72 Internet

eprints.ucm.es <1%

73 Internet

kth.diva-portal.org <1%

74 Internet

repositorio.ug.edu.ec <1%

75 Internet

sidof.segob.gob.mx <1%

76 Internet

www.divulgamat.net <1%

77 Internet

www.fuac.edu.co <1%

78 Internet

www.g-ko.com <1%

79 Internet

www.icmec.org <1%

80 Internet

www.investigo.biblioteca.uvigo.es <1%

81 Internet

www.m-d-n.com <1%

| | | |
|----|-------------|---|
| 82 | Internet | |
| | | www.produccionbovina.com <1% |
| 83 | Internet | |
| | | www.scielo.org.co <1% |
| 84 | Internet | |
| | | www.seimc.org <1% |
| 85 | Internet | |
| | | 1library.co <1% |
| 86 | Publication | |
| | | Carlos Pons Morera. "Análisis crítico del modelo histórico de aplicación de TICs en..." <1% |
| 87 | Publication | |
| | | Cecilia Estrada-Villaseñor, Antonio Rúa, Mercedes Fernández. "Unveiling Subtle (...)" <1% |
| 88 | Publication | |
| | | Daniel Mellado. "<![CDATA[A Security Requirements Engineering Process in Practi..." <1% |
| 89 | Publication | |
| | | Muniz Rivera, Luz Marina. "Analisis sobre metodos de pruebas de hipotesis multi..." <1% |
| 90 | Internet | |
| | | asylumineurope.org <1% |
| 91 | Internet | |
| | | doku.pub <1% |
| 92 | Internet | |
| | | dokumen.pub <1% |
| 93 | Internet | |
| | | es.slideshare.net <1% |
| 94 | Internet | |
| | | idus.us.es <1% |
| 95 | Internet | |
| | | infoscience.epfl.ch <1% |

| | |
|--|-------------|
| 96 | Internet |
| rad.ort.edu.uy | <1% |
| 97 | Internet |
| riull.ull.es | <1% |
| 98 | Internet |
| ruc.udc.es | <1% |
| 99 | Internet |
| seismoverlag.ch | <1% |
| 100 | Internet |
| www.amway.es | <1% |
| 101 | Internet |
| www.grafati.com | <1% |
| 102 | Internet |
| www.infofred.com | <1% |
| 103 | Internet |
| www.proz.com | <1% |
| 104 | Internet |
| www.scribd.com | <1% |
| 105 | Publication |
| "Data Management, Analytics and Innovation", Springer Science and Business Me... | <1% |
| 106 | Publication |
| Benjamín Maraza-Quispe, María Florencia Concha-Fuse, Jorge Luis Torres-Loayza, ... | <1% |
| 107 | Internet |
| www.soloremoto.com | <1% |



Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros
Informáticos



Grado en Ingeniería Informática

Trabajo Fin de Grado

**Desarrollo de un sistema de
retroalimentación explicable basado en
modelos de lenguaje para la plataforma
educativa DeStance en lengua inglesa.**

Autor: Guillermo Palomero Bernal

Tutora: Jelena Bobkina Lukascuka

Cotutora: Elena Domínguez Romero

Madrid, enero – 2026

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: Desarrollo de un sistema de retroalimentación explicable basado en modelos de lenguaje para la plataforma educativa DeStance en lengua inglesa.

Enero – 2026

Autor: Guillermo Palomero Bernal

Tutora: Jelena Bobkina Lukascuka (Departamento de Lingüística Aplicada - UPM)

Cotutora: Elena Domínguez Romero (Departamento de Estudios Ingleses - UCM)

Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

1

Índice general

Resumen

Abstract

| | | |
|----|---|----|
| 46 | 1 Introducción | 1 |
| | 1.1 Punto de partida | 2 |
| | 1.2 Análisis | 3 |
| | 1.2.1 Limitaciones del análisis en De-Stance | 3 |
| | 1.2.2 Limitaciones del módulo ACAIA-FEEDBACK | 3 |
| | 1.2.3 Análisis de los datasets utilizados | 4 |
| | 1.3 Objetivos | 4 |
| | 1.4 Estructura | 5 |
| | 2 Trabajo relacionado y Estado del Arte | 5 |
| | 3 Requisitos | 8 |
| 1 | 3.1 Requisitos funcionales | 9 |
| | 3.1.1 Integración con De-Stance (requisitos heredados) | 9 |
| 10 | 3.1.2 Requisitos (nuevos) | 10 |
| | 3.2 Requisitos no funcionales | 11 |
| | 3.2.1 Rendimiento y disponibilidad | 11 |
| | 3.2.2 Mantenibilidad y extensibilidad | 11 |
| | 3.2.3 Usabilidad y accesibilidad | 12 |
| 1 | 4 Tecnologías usadas | 12 |
| | 4.1 Tecnologías para el Procesamiento del Lenguaje Natural (NLP) | 12 |
| | 4.2 Tecnologías de Desarrollo Web | 13 |
| | Bases de Datos | 14 |
| | 4.3 Lenguajes de Programación | 14 |
| | 4.4 Tecnologías elegidas | 14 |
| | 5 Desarrollo de un sistema de retroalimentación automática aplicada a textos de lengua inglesa | 14 |
| | 5.1 Contexto y diseño del sistema | 14 |
| | 5.2 Textos/Documentación | 16 |
| | 5.2.1 Epistemic words and expressions | 17 |
| | 5.2.2 Effective words and expressions | 18 |
| | 5.2.3 Emotional words and expressions | 19 |
| | 5.2.4 Utilidad | 20 |
| | 5.3 Desarrollo Datasets | 20 |
| | 5.3.1 Primer sistema de procesamiento | 20 |

| | |
|---|----|
| 5.3.2 Segundo sistema de procesamiento | 29 |
| 5.3.3 Tercer sistema de procesamiento | 34 |
| 5.4 Entrenamientos | 35 |
| 5.4.1 Encoder-only Transformer..... | 38 |
| 5.4.2 Encoder-Decoder Transformer | 57 |
| 5.4.3 Entrenamiento BIO..... | 63 |
| 5.5 HUGGING FACE | 64 |
| 5.5.1 App.py | 65 |
| 5.5.2 Importación a DeStance | 69 |
| 5.6 MOOCS..... | 70 |
| 6 Resultados y conclusiones..... | 76 |
| 6.1 Resultados finales | 76 |
| 6.2 Escalabilidad y mantenibilidad | 77 |
| 6.3 Contribución del proyecto..... | 77 |
| 6.4 Desarrollos futuros del proyecto..... | 78 |
| 6.5 Conclusión..... | 79 |
| 6.6 Agradecimientos..... | 79 |
| Bibliografía | 80 |
| ANEXO | 84 |

45

Resumen

El presente Trabajo Fin de Grado se enmarca en el proyecto RACISMMAFF y da continuidad a desarrollos previos realizados en la plataforma web DeStance, que integra ACAIA-FEEDBACK, un módulo específico destinado al análisis lingüístico automático orientado a la recopilación, organización y análisis de textos escritos en diferentes lenguas. El uso de estos recursos en el contexto de un Curso Online Masivo y Abierto (MOOC) integrado en la propia plataforma DeStance permite analizar el desarrollo competencial del alumnado, con especial atención a las competencias interculturales.

A partir de este marco previo, el objetivo principal de este trabajo es reforzar las competencias lingüísticas del estudiantado mediante el apoyo de herramientas tecnológicas avanzadas, promoviendo un enfoque educativo inclusivo, innovador y centrado en el aprendizaje autónomo y autorregulado.

En este contexto, el trabajo se centra en el diseño, implementación y mejora funcional de ACAIA-FEEDBACK, una herramienta automática de análisis de documentos destinada a proporcionar información inmediata sobre el uso de construcciones lingüísticas complejas. Su integración en DeStance permitirá ofrecer al alumnado retroalimentación individualizada y datos estadísticos sobre su producción escrita, favoreciendo una mejora progresiva y sostenida de la competencia comunicativa en contextos académicos.

Desde una perspectiva pedagógica, el feedback proporcionado por ACAIA-FEEDBACK no se concibe únicamente como un apoyo técnico, sino como una herramienta de reflexión sobre el propio uso del lenguaje. Al hacer visibles recursos epistémicos, efectivos y emocionales, el sistema permite que el alumnado tome conciencia de cómo se posiciona discursivamente en sus propios textos, favoreciendo una comprensión crítica del papel que el lenguaje desempeña en la construcción y reproducción de discursos sociales, especialmente en contextos relacionados con el racismo y la discriminación.

Para el desarrollo de la herramienta se utiliza la plataforma Hugging Face, ampliamente empleada para la creación y despliegue de modelos de aprendizaje automático en el ámbito del procesamiento del lenguaje natural. La funcionalidad se integra en la plataforma DeStance mediante un servicio externo conectado al sistema a través del back-end, desde el cual se gestionan las peticiones necesarias para que los resultados sean finalmente visualizados en el front-end de la aplicación.

1 Abstract

95 This Final Degree Project is framed within the RACISMMAFF project and builds upon previous developments carried out on the DeStance web platform, which integrates ACAIA-FEEDBACK, a specific module designed for automatic linguistic analysis aimed at the collection, organization, and analysis of written texts in different languages. The use of these resources within the context of a Massive Open Online Course (MOOC) integrated into the DeStance platform enables the analysis of students' competence development, with particular attention to intercultural competences.

39 Based on this prior framework, the main objective of this project is to strengthen 43 students' linguistic competences through the use of advanced technological tools, promoting an inclusive and innovative educational approach focused on autonomous and self-regulated learning.

1 Within this context, the project focuses on the design, implementation, and 1 functional enhancement of ACAIA-FEEDBACK, an automatic document analysis tool intended to provide immediate information on the use of complex linguistic constructions. Its integration into DeStance allows for the provision of individualized feedback and statistical data on students' written production, fostering a progressive and sustained improvement of communicative competence in academic contexts.

99 From a pedagogical perspective, the feedback provided by ACAIA-FEEDBACK is not conceived merely as technical support, but as a tool for reflection on language use. By making epistemic, affective, and emotional resources visible, the system enables students to become aware of how they position themselves discursively in their own texts, thereby promoting a critical understanding of the role language plays in the construction and reproduction of social discourses, particularly in contexts related to racism and discrimination.

43 105 For the development of the tool, the Hugging Face platform is employed, which 59 is widely used for the creation and deployment of machine learning models in the field of Natural Language Processing. The functionality is integrated into the DeStance platform through an external service connected to the system via the back-end, from which the necessary requests are managed so that the results are ultimately visualized in the application's front-end.

1 Introducción

En la actualidad, en un contexto europeo marcado por el aumento de discursos racistas y xenófobos, tal y como reflejan informes institucionales recientes [1], surge el proyecto RACISMMAFF con el objetivo de analizar cómo las personas se posicionan discursivamente en textos relacionados con la inmigración y el racismo [2]. El proyecto busca contribuir a la comprensión de las prácticas discursivas que normalizan la desigualdad y, al mismo tiempo, favorecer una evolución positiva del discurso del estudiantado universitario y del profesorado en formación, promoviendo el pensamiento crítico y la empatía.

Con el fin de trasladar estos objetivos al ámbito educativo, se desarrolló la plataforma DeStance, una aplicación web orientada a la recopilación, organización y análisis de textos producidos en lengua inglesa, con previsión de expansión a otras lenguas en el futuro [3]. En este contexto se integra ACAIA-FEEDBACK, una herramienta automatizada de análisis lingüístico desarrollada a partir de tecnologías de procesamiento del lenguaje natural disponibles en la plataforma Hugging Face, cuya funcionalidad permite conectar el procesamiento de los textos con DeStance y presentar los resultados del análisis de forma accesible al usuario final.

Este proyecto no parte de cero, sino que se inserta en una línea de trabajo ya consolidada y desarrollada por mis compañeros Ana Clara Murillo García [4], Daniel Ramón Robertson [5] y Gonzalo Álvarez García [6]. El presente Trabajo Fin de Grado da continuidad a dicho desarrollo, incorporando mejoras tanto a nivel funcional como visual, con el objetivo de optimizar la experiencia de uso de la plataforma. De este modo, se pretende facilitar al alumnado el acceso a los resultados del análisis lingüístico y, al mismo tiempo, apoyar al profesorado reduciendo la necesidad de realizar análisis manuales de los textos, favoreciendo así una gestión más eficiente del proceso de evaluación.



1.1 Punto de partida

Como se ha indicado previamente, este Trabajo Fin de Grado no parte de cero, sino que se apoya en un desarrollo previo ya existente. En concreto, la estructura principal del proyecto se encontraba ya definida, incluyendo tanto la plataforma DeStance como la herramienta ACAIA-FEEDBACK.

En la Figura 1.1.1 se muestra el estado inicial de la plataforma DeStance al comienzo de este Trabajo Fin de Grado.

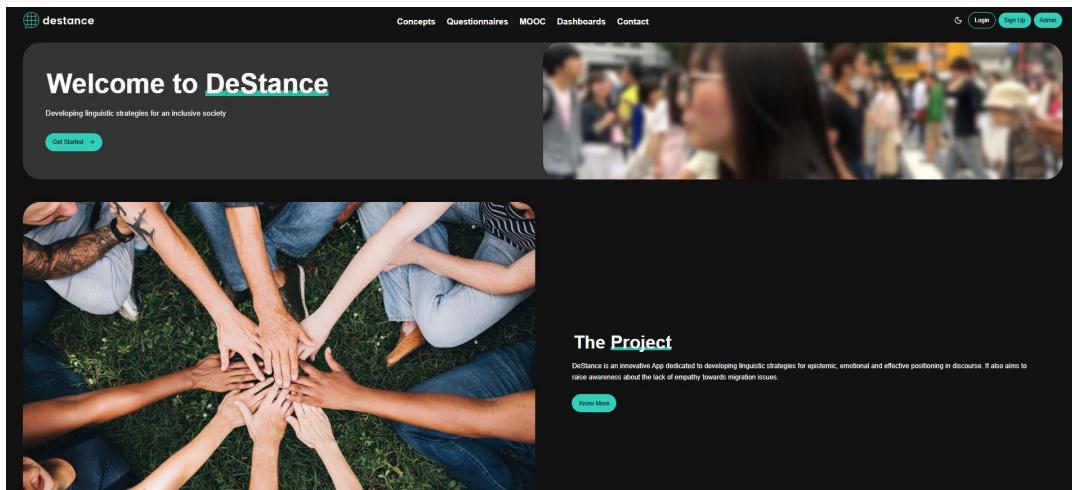


Figura 1.1.1 Plataforma DeStance estado inicial del TFG

Tal y como se observa en la figura 1.1.1, la plataforma DeStance ya contaba con una estructura funcional definida, así como con una interfaz orientada al uso educativo.

Además de la plataforma DeStance, el proyecto contaba con una herramienta destinada a la generación de retroalimentación automática, denominada ACAIA-FEEDBACK.

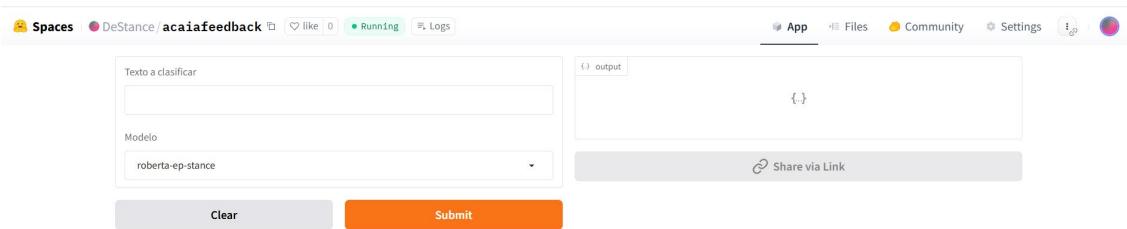


Figura 1.1.2 Herramienta ACAIA-FEEDBACK estado inicial

En la Figura 1.1.2 se muestra el estado inicial de la herramienta ACAIA-FEEDBACK al comienzo de este Trabajo Fin de Grado.

15 Gracias a la labor de las tutoras Elena Domínguez Romero y Jelena Bobkina, ha sido posible acceder al contexto actual del proyecto y a la documentación necesaria para comprender su estado de desarrollo. A partir de este punto, se han identificado las mejoras requeridas, tanto a nivel funcional como visual, que han servido de base para la planificación y ejecución del presente Trabajo Fin de Grado.

1.2 Análisis

1 El análisis realizado en trabajos previos sobre la plataforma **De-Stance** y el módulo **ACAIA-FEEDBACK** puso de manifiesto una serie de limitaciones que afectaban tanto al rendimiento de los modelos utilizados como a la calidad del análisis discursivo proporcionado. Estas limitaciones se relacionan principalmente con dos factores: por un lado, la elección y configuración de los modelos de lenguaje empleados; por otro, la calidad y consistencia de los conjuntos de datos utilizados durante su entrenamiento y evaluación.

1.2.1 Limitaciones del análisis en De-Stance

En la plataforma De-Stance, el análisis de textos estaba implementado exclusivamente mediante modelos basados en la arquitectura T5. Aunque este enfoque permitía generar anotaciones automáticas, en la práctica se observaban problemas recurrentes en la salida del sistema. En particular, aparecían inconsistencias en el etiquetado, así como ausencias de etiquetas esperadas en determinados fragmentos del texto analizado.

Adicionalmente, el sistema no ofrecía métricas de confianza asociadas a las etiquetas generadas, lo que dificultaba evaluar la fiabilidad de los resultados y limitaba su interpretabilidad en un contexto educativo.

1.2.2 Limitaciones del módulo ACAIA-FEEDBACK

Por su parte, el módulo ACAIA-FEEDBACK incorporaba modelos basados en la arquitectura RoBERTa, orientados a la clasificación de expresiones discursivas. Sin embargo, en su estado inicial, estos modelos presentaban una complejidad limitada y no alcanzaban un rendimiento satisfactorio en la detección y clasificación de expresiones epistémicas, efectivas y emocionales, especialmente en textos largos o con estructuras discursivas variadas.

Asimismo, dentro de ACAIA-FEEDBACK también se empleaban modelos basados en T5, que en general ofrecían un comportamiento más estable. No obstante, estos modelos seguían presentando deficiencias relevantes, como errores de clasificación, ausencia de determinadas etiquetas y dificultades para generalizar ante ejemplos menos frecuentes o combinaciones complejas de marcadores.

Además de las limitaciones técnicas, la versión inicial del sistema mostraba también una presentación poco interpretativa de los resultados, lo que dificultaba su uso por parte de estudiantes y docentes y reducía el valor pedagógico de la retroalimentación generada.

1.2.3 Análisis de los datasets utilizados

El análisis conjunto de estas limitaciones permitió identificar un factor común subyacente: la inconsistencia y falta de estructuración adecuada de los datasets utilizados durante los entrenamientos.

En particular, la ausencia de ejemplos negativos, la falta de variedad en las combinaciones de etiquetas y la presencia de anotaciones poco homogéneas afectaban directamente a la capacidad de los modelos para aprender patrones robustos y generalizar correctamente. Como consecuencia, los modelos tendían a producir salidas incompletas o sesgadas, especialmente en situaciones donde el texto no se ajustaba a los patrones más frecuentes del entrenamiento.

1.3 Objetivos

A partir del análisis del estado inicial de la plataforma DeStance y del módulo ACAIA-FEEDBACK, en el que se han identificado limitaciones tanto a nivel técnico como en la interpretación de los resultados, se definen a continuación los objetivos del presente Trabajo Fin de Grado.

El objetivo principal de este trabajo es la reestructuración e implementación de una nueva versión de ACAIA-FEEDBACK que permita mejorar la calidad, fiabilidad e interpretabilidad del análisis lingüístico proporcionado. Para ello, se plantea la actualización y optimización de los modelos de procesamiento del lenguaje natural empleados, así como la incorporación de métricas cuantitativas e indicadores de confianza que faciliten la evaluación de los resultados obtenidos.

Asimismo, se persigue mejorar la integración de la herramienta en la plataforma DeStance y optimizar la visualización de los resultados generados, con el fin de ofrecer una presentación más clara, informativa y útil tanto para el alumnado como para el profesorado, reforzando así el valor pedagógico de la retroalimentación automática.

Objetivos específicos

Para alcanzar el objetivo general planteado, se definen los siguientes objetivos específicos:

1. Revisar, corregir y optimizar el sistema de preprocesamiento de los datos utilizados, garantizando una mayor consistencia y calidad de la información de entrada para los modelos de procesamiento del lenguaje natural.
2. Diseñar e implementar un nuevo sistema de clasificación basado en la arquitectura RoBERTa, adaptado a las nuevas necesidades del proyecto, e integrarlo en la plataforma ACAIA-FEEDBACK.
3. Evaluar y comparar distintos enfoques y modelos de clasificación alternativos, con el fin de analizar su adecuación y rendimiento sobre los nuevos conjuntos de datos.
4. Mejorar el rendimiento de los modelos basados en la arquitectura T5 empleados en ACAIA-FEEDBACK, optimizando su comportamiento en la generación de anotaciones lingüísticas.
5. Incorporar sistemas de puntuación de confianza asociados a las etiquetas generadas, otorgando una fiabilidad de los resultados proporcionados por la herramienta.
6. Mejorar la visualización y presentación de los resultados del análisis lingüístico en la plataforma DeStance, proporcionando información

contextual clara y comprensible que incremente su valor pedagógico para estudiantes y docentes.

1.4 Estructura

La presente memoria se organiza en siete capítulos, además del resumen y el abstract, con el objetivo de describir de forma progresiva y coherente el contexto, el análisis, el desarrollo técnico y los resultados del Trabajo de Fin de Grado.

3 En el **Capítulo 1** se introduce el contexto general del proyecto, se describe el punto de partida a partir de desarrollos previos, se expone un análisis detallado de las limitaciones detectadas en el estado inicial de la plataforma De-Stance y del módulo ACAIA-FEEDBACK, así como de los conjuntos de datos utilizados, y se establecen los objetivos que guían el trabajo realizado.

89 1 20 El **Capítulo 2** presenta el trabajo relacionado y el estado del arte, analizando distintas tecnologías y enfoques existentes en el ámbito de la retroalimentación automática y el procesamiento del lenguaje natural. Este análisis permite justificar las decisiones técnicas adoptadas posteriormente, en particular la elección de plataformas y modelos.

35 1 El **Capítulo 3** recoge la especificación de los requisitos funcionales y no funcionales que afectan directamente a los componentes desarrollados o ampliados en este trabajo, incluyendo la integración con De-Stance, la evaluación y retroalimentación automática, y el tratamiento de los datasets.

1 En el **Capítulo 4** se describen las principales tecnologías empleadas durante el desarrollo del proyecto, tanto en lo relativo al procesamiento del lenguaje natural como a la infraestructura y herramientas utilizadas para el entrenamiento y despliegue de los modelos.

El **Capítulo 5** constituye el núcleo del trabajo y detalla el desarrollo del sistema de retroalimentación multilingüe. En él se aborda el estudio de la documentación lingüística, la generación y tratamiento de los datasets, los distintos enfoques de entrenamiento de modelos Transformer, su optimización y despliegue mediante Hugging Face, así como la integración final del sistema en la plataforma De-Stance y en los módulos MOOC.

Por último, el **Capítulo 6** presenta los resultados y conclusiones del proyecto, incluyendo una evaluación global del sistema desarrollado, la contribución del trabajo al proyecto De-Stance, las posibles líneas de desarrollo futuro y una reflexión final sobre la experiencia y el aprendizaje adquiridos.

2 Trabajo relacionado y Estado del Arte

En el marco de los trabajos previos a Trabajo de fin de grado se realizaron unos estudios relacionados con la retroalimentación y el análisis automático de textos escritos, se analizaron diversas tecnologías ampliamente utilizadas en el ámbito del procesamiento del lenguaje natural y la asistencia a la

escritura. Entre ellas se encuentran herramientas como Grammarly [7], ProWritingAid [8] y QuillBot [9], cuyo objetivo principal es la corrección gramatical, la mejora estilística y la reformulación de textos. Aunque estas herramientas resultan eficaces como asistentes de escritura, su enfoque está orientado a la edición y mejora superficial del texto, sin proporcionar un análisis explícito del posicionamiento epistémico, afectivo o emocional del discurso, ni permitir la identificación y etiquetación de segmentos concretos desde una perspectiva discursiva.

50 En el ámbito académico, diversos trabajos recientes han abordado el uso de técnicas de procesamiento del lenguaje natural para la provisión de retroalimentación automática en contextos educativos. Estudios como el de Ranalli analizan el potencial de los sistemas de retroalimentación automática para apoyar la escritura en segundas lenguas, subrayando la importancia de que dichos sistemas no se limiten a la corrección formal [10], sino que ofrezcan información interpretable y pedagógicamente relevante. Estos enfoques ponen de manifiesto la necesidad de sistemas capaces de analizar el contenido discursivo del texto y no únicamente su forma superficial.

1 Asimismo, se consideró el uso de modelos de lenguaje de gran escala como GPT-4 [11], que destacan por su capacidad para generar texto y ofrecer retroalimentación flexible y personalizada. No obstante, este tipo de modelos se presenta como una solución generalista y de carácter opaco, lo que dificulta el control sobre el proceso de inferencia y la reproducibilidad experimental. Además, su uso no permite integrar modelos entrenados específicamente para tareas concretas, como la identificación de posicionamientos discursivos, lo que limita su adecuación para un entorno académico con objetivos analíticos bien definidos.

Por otra parte, plataformas de análisis textual como Voyant Tools [12] ofrecen métricas cuantitativas útiles para la exploración de textos y corpus, como la frecuencia léxica o la legibilidad. Sin embargo, estas herramientas se centran en análisis globales y estadísticos, y no contemplan el análisis semántico profundo ni la detección automática de posicionamientos discursivos a nivel de segmento, aspectos fundamentales para el propósito del presente trabajo.

En conjunto, el análisis de estas tecnologías puso de manifiesto que, si bien cada una resulta adecuada en su ámbito de aplicación específico, ninguna permite abordar de forma integrada la identificación automática de posicionamientos epistémicos en textos escritos, manteniendo al mismo tiempo control sobre los modelos utilizados y garantizando la reproducibilidad de los resultados. Esta limitación motivó la necesidad de recurrir a soluciones que permitieran el despliegue de modelos propios, entrenados específicamente para esta tarea, mediante plataformas especializadas alineadas con las prácticas habituales del estado del arte en investigación en procesamiento del lenguaje natural.

12 5 En este contexto, y dentro de la línea de investigación en la que se enmarca el presente trabajo, se llevó a cabo un estudio previo de distintas alternativas tecnológicas para el despliegue y acceso a modelos de procesamiento del lenguaje natural orientados a la identificación automática de posicionamientos epistémicos en textos escritos. Dicho estudio tuvo como objetivo principal encontrar una solución que permitiera integrar modelos basados en

arquitecturas *transformer* en una aplicación web, garantizando su accesibilidad, reproducibilidad y viabilidad técnica dentro de un entorno académico.

Entre las alternativas analizadas se contempló, en primer lugar, la posibilidad de ejecutar los modelos en infraestructura propia o entornos locales. Esta opción, si bien ofrece un alto grado de control sobre el sistema, presentaba importantes inconvenientes, como la necesidad de disponer de recursos computacionales avanzados, especialmente GPU, así como de asumir tareas de configuración, mantenimiento y escalado de la infraestructura. Estas exigencias suponían una carga técnica elevada y poco compatible con los objetivos y el alcance de un trabajo de carácter académico y experimental.

Asimismo, se evaluaron plataformas de computación en la nube de carácter generalista, que permiten desplegar aplicaciones y servicios de inteligencia artificial con gran flexibilidad. No obstante, este tipo de soluciones requiere una configuración técnica compleja, incluyendo la gestión de instancias, recursos y mecanismos de seguridad. Además, al no estar específicamente orientadas al procesamiento del lenguaje natural, estas plataformas dificultan la iteración rápida sobre modelos y experimentos, lo que puede desviar el foco del proyecto hacia aspectos infraestructurales ajenos a la investigación lingüística.

Por otra parte, también se consideró el uso de APIs comerciales de procesamiento del lenguaje natural, que ofrecen acceso inmediato a modelos preentrenados y facilitan su integración en aplicaciones externas. Sin embargo, estas soluciones presentan limitaciones significativas desde una perspectiva académica, ya que no permiten el despliegue de modelos propios ni proporcionan transparencia sobre los procesos de entrenamiento y funcionamiento interno. En consecuencia, su uso limita la reproducibilidad de los experimentos y reduce la capacidad de análisis crítico de los resultados obtenidos.

Como resultado de este análisis previo, se optó por utilizar Hugging Face Space, una plataforma especializada que permite desplegar modelos entrenados y acceder a ellos de forma sencilla mediante interfaces reutilizables [13]. Esta plataforma destaca por su amplia adopción en la comunidad investigadora en procesamiento del lenguaje natural, así como por su idoneidad para entornos académicos, al facilitar la experimentación, la reproducibilidad y la difusión de los modelos sin introducir una complejidad infraestructural innecesaria.

En consecuencia, la selección de esta plataforma se entiende como el resultado de una decisión técnica fundamentada en estudios previos, alineada con el estado del arte y coherente con los objetivos del presente trabajo, permitiendo centrar los esfuerzos en el análisis del posicionamiento discursivo y en la aplicación práctica de los modelos desarrollados.

Desde una perspectiva académica, diversos trabajos han analizado el impacto de la retroalimentación correctiva escrita automatizada en contextos educativos, subrayando que la eficacia del feedback no depende únicamente

de su disponibilidad, sino de su grado de especificidad, claridad y adecuación cognitiva. En particular, estudios sobre *automated written corrective feedback* han mostrado que el feedback genérico tiende a aumentar la carga cognitiva y a reducir la tasa de correcciones exitosas, mientras que el feedback específico y localizado favorece un uso más efectivo por parte del alumnado. Asimismo, investigaciones en adquisición de segundas lenguas han puesto de relieve la importancia del procesamiento del feedback y de las condiciones en las que este tiene lugar, destacando la influencia del medio de composición y del diseño de las tareas de procesamiento en la calidad de los textos revisados [14]. Estos trabajos refuerzan la necesidad de sistemas capaces de identificar y etiquetar segmentos discursivos concretos para ofrecer retroalimentación interpretable y pedagógicamente relevante, como la propuesta desarrollada en el presente trabajo.

1 En esta línea, estudios recientes en tecnología educativa han analizado el uso de técnicas de *natural language processing* para proporcionar retroalimentación automática sobre la escritura argumentativa en contextos universitarios, mostrando que el diseño del feedback —y no solo su automatización— resulta determinante para mejorar la calidad argumentativa de los textos [15]. Estos trabajos, orientados a escenarios escalables e independientes del docente, refuerzan la pertinencia de enfoques basados en análisis discursivo a nivel de segmento, como los adoptados en el presente trabajo.

78 A partir de estos trabajos y herramientas previas, ACAIA-FEEDBACK retoma la idea de la retroalimentación automática como apoyo a la escritura académica, compartiendo con estos enfoques el objetivo de mejorar la calidad de las producciones escritas del alumnado. No obstante, frente a soluciones centradas en la corrección formal o en métricas globales del texto, ACAIA-FEEDBACK introduce un enfoque diferencial basado en el análisis discursivo a nivel de segmento, orientado específicamente a la identificación automática de posicionamientos epistémicos, afectivos y emocionales. Asimismo, a diferencia de modelos generalistas y opacos, la herramienta se fundamenta en modelos propios entrenados para tareas concretas, lo que permite un mayor control sobre el proceso de análisis, garantiza la reproducibilidad de los resultados y facilita la interpretación pedagógica del feedback generado. De este modo, ACAIA-FEEDBACK se sitúa como una contribución relevante en el ámbito de la retroalimentación automática, al integrar avances del procesamiento del lenguaje natural con un diseño pedagógico orientado a contextos educativos reales y escalables.

66

1 3 Requisitos

47 En este capítulo se especifican los requisitos funcionales y no funcionales que afectan directamente a los componentes desarrollados o ampliados en el presente trabajo, en particular al tratamiento de los datasets, a la mejora del módulo ACAIA-FEEDBACK y a su integración en la plataforma De-Stance. La identificación de estos requisitos se llevó a cabo a partir del análisis de los desarrollos previos de la plataforma, de la experiencia acumulada en trabajos

anteriores relacionados con ACAIA-FEEDBACK y de un estudio preliminar del funcionamiento del sistema en contextos educativos. Asimismo, se tuvieron en cuenta los objetivos específicos del presente Trabajo Fin de Grado y las limitaciones propias de un entorno académico y experimental. La priorización de los requisitos se estableció en función de su impacto en la funcionalidad del sistema, su relevancia para garantizar una integración efectiva con De-Stance y su contribución a la calidad y utilidad pedagógica de la retroalimentación generada.

3.1 Requisitos funcionales

3.1.1 Integración con De-Stance (requisitos heredados)

- **Integración con la plataforma De-Stance.**
Prioridad: Alta.

El módulo ACAIA-FEEDBACK debe integrarse en De-Stance, permitiendo a los usuarios acceder a la funcionalidad de retroalimentación sin necesidad de instalar software adicional.

- **Consistencia visual y funcional.**
Prioridad: Alta.

El módulo debe mantener la coherencia visual y funcional con el resto de la plataforma, reutilizando los mecanismos de autenticación y roles de usuario existentes.

- **Comunicación mediante API.**
Prioridad: Alta.

La comunicación entre De-Stance y el módulo ACAIA-FEEDBACK debe realizarse mediante una API que permita el envío de textos y la recepción de los resultados del análisis.

- **Persistencia de resultados.**
Prioridad: Alta.

Los textos analizados y la retroalimentación generada deben almacenarse en la plataforma para permitir su consulta posterior y su uso en procesos de evaluación y seguimiento.

- **Procesamiento de textos en inglés.**
Prioridad: Alta.

El sistema debe aceptar textos escritos en inglés y analizarlos automáticamente.

- **Identificación de posicionamientos discursivos.**
Prioridad: Alta.

ACAIA-FEEDBACK debe detectar y clasificar expresiones epistémicas, afectivas y emocionales presentes en el texto, a nivel de segmento.

- **Mejora del rendimiento del análisis.**

Prioridad: Alta.

El sistema debe mejorar la precisión y la consistencia de la clasificación respecto a versiones previas del módulo, reduciendo errores de etiquetado y ausencias de clasificación.

- **Generación de retroalimentación pedagógica.**

Prioridad: Alta.

Los resultados del análisis deben transformarse en retroalimentación comprensible para el estudiante, orientada a mejorar la claridad, el tono y la argumentación del texto.

3.1.2 Requisitos (nuevos)

- **Consistencia de los datos de entrada.**

Prioridad: Alta.

Todos los datos procesados por el sistema deben cumplir criterios de consistencia y normalización, garantizando una estructura homogénea y adecuada para el entrenamiento y evaluación de los modelos de procesamiento del lenguaje natural.

- **Balance entre categorías.**

Prioridad: Alta.

Los datasets utilizados para el entrenamiento y evaluación de los modelos deben minimizar el desbalance entre clases de posicionamiento.

- **Mejora en los datasets.**

Prioridad: Alta.

El proceso de limpieza, normalización y preparación de los datasets debe ser mejorado y documentado.

- **Posibilidad de ampliación.**

Prioridad: Media.

El sistema debe permitir la incorporación futura de nuevos datos anotados sin comprometer la coherencia del conjunto existente.

- **Detección de posicionamientos emocionales (Emo).**

Prioridad: Alta.

ACAIA-FEEDBACK debe incorporar la detección automática de posicionamientos emocionales (Emo), tanto a nivel de análisis lingüístico como de visualización de resultados en la plataforma DeStance.

- **Mejora de la retroalimentación en DeStance.**

Prioridad: Alta.

El sistema debe mejorar la calidad y la presentación visual de la retroalimentación proporcionada en la plataforma DeStance, facilitando su interpretación por parte del alumnado y del profesorado.

- **Investigar más sistemas de modelo para mejorar el rendimiento**
Prioridad: Alta.

Investigar mas modelos de entrenamiento para ver si se puede obtener mejores resultados

3.2 Requisitos no funcionales

3.2.1 Rendimiento y disponibilidad

- **Tiempo de respuesta.**
Prioridad: Alta.

El sistema debe proporcionar los resultados del análisis en un tiempo compatible con su uso interactivo en entornos educativos.

- **Disponibilidad.**
Prioridad: Alta.

ACAIA-FEEDBACK debe estar disponible de forma continua, siguiendo las condiciones de disponibilidad establecidas por la plataforma De-Stance.

3.2.2 Mantenibilidad y extensibilidad

- **Modularidad.**
Prioridad: Alta.

El sistema debe estar estructurado de forma modular para facilitar la sustitución o mejora de los modelos de lenguaje y el tratamiento de los datasets.

- **Extensibilidad.**
Prioridad: Media.

La arquitectura debe permitir la incorporación futura de nuevos modelos o idiomas sin necesidad de reestructurar el sistema completo.

3.2.3 Usabilidad y accesibilidad

- **Interfaz clara e intuitiva.**
Prioridad: Alta.

La interfaz debe permitir a los estudiantes interpretar fácilmente la retroalimentación proporcionada.

- **Accesibilidad.**
Prioridad: Alta.

La aplicación debe cumplir con criterios básicos de accesibilidad web, siguiendo las directrices WCAG 2.1 AA.

Estos requisitos han servido como marco de referencia para el diseño y desarrollo de las soluciones presentadas en los capítulos posteriores, guiando tanto el tratamiento de los datasets como la mejora del módulo ACAIA-FEEDBACK y su integración efectiva en la plataforma De-Stance.

4 Tecnologías usadas

La selección de las tecnologías empleadas en este trabajo no ha sido arbitraria, sino que responde tanto a los requisitos técnicos y académicos del proyecto como a las decisiones adoptadas en trabajos previos desarrollados en el marco de la plataforma DeStance y del módulo ACAIA-FEEDBACK.

4.1 Tecnologías para el Procesamiento del Lenguaje Natural (NLP)

Hugging Face Transformers

La biblioteca Transformers de Hugging Face [16] constituye el núcleo del procesamiento del lenguaje natural en este proyecto. Esta biblioteca proporciona implementaciones optimizadas de modelos de lenguaje de última generación basados en arquitecturas *Transformer*, como RoBERTa, DeBERTa y T5, ampliamente utilizados en tareas de clasificación, etiquetado y generación de texto.

La biblioteca admite una integración nativa con los principales frameworks de *deep learning*, como PyTorch [17], TensorFlow [18] y JAX [19], lo que facilita la portabilidad de los modelos entre entornos de entrenamiento e inferencia. Asimismo, cada arquitectura se define en módulos independientes, lo que permite su personalización y adaptación a tareas específicas de investigación.

Adicionalmente, la biblioteca Transformers proporciona abstracciones específicas para tareas de clasificación tanto a nivel de secuencia como a nivel de token, lo que ha permitido implementar distintos enfoques de análisis lingüístico dentro de un mismo marco tecnológico. Esta flexibilidad ha sido especialmente relevante para el desarrollo de estrategias de entrenamiento diferenciadas, incluyendo modelos de clasificación multietiqueta y esquemas de etiquetado secuencial basados en BIO [20].

Hugging Face Spaces

Los modelos desarrollados se han desplegado mediante Hugging Face Spaces [21], una plataforma que permite alojar aplicaciones de aprendizaje automático y exponerlas a través de APIs accesibles. Esta solución ha resultado especialmente adecuada para este proyecto, ya que facilita la integración directa de los modelos con la plataforma web De-Stance, sin necesidad de gestionar infraestructuras propias.

El uso de Spaces permite desacoplar la lógica de análisis del *front-end*, mejorando la escalabilidad y mantenibilidad del sistema.

Optuna

Para la optimización de hiperparámetros en los modelos *encoder-only*, se ha utilizado Optuna [22], una librería de optimización automática que permite explorar de forma eficiente el espacio de hiperparámetros. A diferencia de enfoques exhaustivos como *Grid Search*, Optuna emplea estrategias adaptativas que reducen el coste computacional y priorizan las configuraciones con mejor rendimiento.

Su integración en el proceso experimental ha permitido mejorar los resultados obtenidos en términos de *F1 Macro*, manteniendo un equilibrio entre rendimiento y tiempo de entrenamiento.

Google Colab

Durante la fase de entrenamiento y experimentación, se ha utilizado Google Colab [23] como entorno de ejecución. Esta plataforma proporciona acceso a recursos computacionales con GPU de forma sencilla, lo que ha permitido acelerar los entrenamientos y realizar pruebas con distintos modelos y configuraciones sin necesidad de infraestructuras locales especializadas.

4.2 Tecnologías de Desarrollo Web

JavaScript y Svelte

La interfaz web de la plataforma DeStance está desarrollada utilizando JavaScript [24] y el framework Svelte [25], un entorno moderno para la creación de interfaces reactivas. Svelte destaca por su enfoque basado en compilación, lo que permite generar aplicaciones ligeras y eficientes, mejorando el rendimiento en tiempo de ejecución.

La comunicación entre el *front-end* y los modelos de análisis se realiza mediante peticiones HTTP a las APIs expuestas en Hugging Face, integrando de forma transparente el procesamiento del lenguaje natural en la experiencia del usuario.

Bases de Datos

MongoDB

1 Para el almacenamiento de los resultados del análisis, se ha empleado MongoDB [26], una base de datos orientada a documentos que ofrece una elevada flexibilidad y escalabilidad. Su modelo de datos resulta especialmente adecuado para almacenar estructuras heterogéneas, como etiquetas, métricas y resultados de análisis lingüístico, facilitando su posterior consulta y visualización.

4.3 Lenguajes de Programación

Python

1 Python [27] ha sido el lenguaje principal utilizado en el desarrollo del proyecto, tanto para el entrenamiento de modelos como para su despliegue. Su amplio ecosistema de librerías orientadas al aprendizaje automático y al procesamiento del lenguaje natural lo convierten en una elección idónea para este tipo de aplicaciones.

JavaScript

1 JavaScript [24] se ha utilizado en el desarrollo del *front-end* de la aplicación web, permitiendo la interacción del usuario con el sistema y la visualización de los resultados de forma dinámica.

4.4 Tecnologías elegidas

En conjunto, las tecnologías seleccionadas permiten construir un sistema modular, escalable y fácilmente extensible, integrando modelos avanzados de procesamiento del lenguaje natural en una plataforma web real. Esta combinación facilita tanto la experimentación académica como la aplicación práctica del sistema en contextos educativos, alineándose con los objetivos del proyecto DeStance y del marco de investigación RACISMMAFF.

5 Desarrollo de un sistema de retroalimentación automática aplicada a textos de lengua inglesa

5.1 Contexto y diseño del sistema

Una vez definido el alcance del presente Trabajo de Fin de Grado, fue necesario contextualizar el estado inicial del sistema a partir del cual se abordó su desarrollo. En el momento de inicio del proyecto, la plataforma De-Stance y el

módulo ACAIA-FEEDBACK contaban con una implementación funcional que permitía el análisis automático de textos, si bien presentaban diversas limitaciones tanto a nivel de rendimiento como de interpretabilidad de los resultados.

9 Gracias a dicho trabajo previo y a la orientación de las tutoras, se dispuso desde el inicio de un sistema funcional sobre el que poder trabajar. En esta fase inicial, el esfuerzo se centró en la revisión y comprensión de la aplicación web existente, así como en el estudio detallado de la memoria y la documentación asociada a los proyectos ACAIA y De-Stance. Este proceso permitió adquirir una visión global del sistema y delimitar con precisión el alcance de las mejoras y desarrollos abordados en este trabajo.

1 El diseño del sistema se basa en el uso de arquitecturas Transformer, adoptando una separación funcional entre modelos *encoder-only* y *encoder-decoder* en función de la naturaleza de cada tarea. Esta decisión responde tanto a criterios teóricos como a criterios de ingeniería. Por un lado, los modelos *encoder-only* de estilo BERT están optimizados para tareas discriminativas como clasificación o etiquetado, donde el objetivo es asignar una etiqueta a un segmento o a cada token a partir de representaciones contextuales bidireccionales; en este escenario, estos modelos tienden a ofrecer un equilibrio favorable entre precisión, estabilidad y coste computacional, al realizar una única pasada de codificación y evitar la decodificación autoregresiva [28].

3 Por otro lado, los modelos *encoder-decoder*, como T5 o BART, están diseñados para tareas generativas y de transformación de secuencias, donde la salida del modelo es una secuencia estructurada de tokens; en estas tareas, la capacidad de generar texto condicionado resulta más adecuada para producir salidas con formato controlado (p. ej., inserción de anotaciones o reformulación), aunque ello implique un mayor coste computacional y una mayor latencia debido a la decodificación token a token [29], [30].

En este sentido, el presente trabajo mantiene una separación arquitectónica coherente con el objetivo de cada subtarea: clasificación/etiquetado cuando se requiere decisión rápida y precisa sobre segmentos o tokens; y generación/inserción de texto anotado cuando la salida debe ser una secuencia explícita. Esta elección permite además mejorar la mantenibilidad del sistema, ya que cada componente puede evaluarse y actualizarse de forma independiente, favoreciendo la reproducibilidad experimental.

En el marco de este diseño, el sistema aborda distintas categorías de posicionamiento discursivo, entre las que se incluyen posicionamientos epistémicos, afectivos y emocionales, cada una de ellas con características lingüísticas y estructurales diferenciadas. Mientras que las categorías epistémica y afectiva se modelan principalmente a nivel de segmento o enunciado, la categoría emocional (EMO) presenta particularidades tanto a nivel lingüístico como a nivel de disponibilidad de datos anotados.

Durante las fases iniciales de experimentación se observó que, debido a la menor cantidad de ejemplos disponibles para la categoría EMO, los enfoques basados en modelos independientes y decisiones discretas producían resultados menos estables y precisos que en el caso de las otras categorías. Esta limitación motivó la adopción de un enfoque alternativo que permitiera aprovechar mejor

la información contextual del texto y reducir la fragmentación del proceso de análisis.

A partir de esta consideración, se introduce un enfoque específico para la categoría EMO basado en el uso de un esquema de etiquetado BIO a nivel de token, que permite realizar una ejecución continua del análisis y capturar señales distribuidas a lo largo del texto. Este enfoque resulta especialmente adecuado en escenarios con datos limitados, ya que incrementa el número de instancias de aprendizaje y favorece una mayor coherencia en la detección de expresiones emocionales.

Es importante destacar que el uso del esquema BIO (Begin–Inside–Outside) no implica el empleo de arquitecturas encoder–decoder, ya que BIO define exclusivamente el formato de las etiquetas en una tarea de sequence labeling [31]. En este contexto, la categoría EMO se aborda mediante modelos encoder-only, que generan representaciones contextuales por token y predicen directamente las etiquetas BIO correspondientes en una única ejecución, evitando la necesidad de combinar resultados procedentes de múltiples modelos. Para las categorías epistémica y afectiva, cuya interpretación depende de unidades discursivas de mayor extensión y para las que se dispone de conjuntos de datos más equilibrados, se mantienen enfoques de clasificación a nivel de segmento.

5.2 Textos/Documentación

El primer paso en el desarrollo de este Trabajo de Fin de Grado consistió en el estudio detallado de la documentación proporcionada por las tutoras del proyecto. Dicha documentación está compuesta por tres documentos principales en formato Word, elaborados en el marco del proyecto RACISMMAFF por Marta Carretero y Elena Domínguez, responsables del mismo, y utilizados como material base en la plataforma De-Stance. Estos documentos describen un sistema de anotación lingüística diseñado para el análisis del posicionamiento discursivo en textos escritos.

Estos documentos —*Epistemic words and expressions*, *Effective words and expressions* y *Emotional words and expressions*— constituyen la referencia teórica y operativa sobre la que se fundamenta tanto el proceso de anotación manual del corpus como el diseño de los conjuntos de datos empleados en este trabajo. Su uso ha sido autorizado y guiado por las tutoras responsables del proyecto.

En esta memoria se adopta una terminología unificada para referirse a las distintas dimensiones del posicionamiento discursivo. En concreto, el término **postura epistémica** se utiliza como traducción de *epistemic stance*; el término **postura efectiva** como traducción de *effective stance*, entendida como la orientación del discurso a influir en acciones, actitudes o decisiones; y el término **postura emocional** como traducción de *emotional stance*, relacionada con la expresión y evaluación de emociones. A lo largo del documento, estos términos se emplean de manera consistente.

Este sistema de anotación distingue tres grandes tipos de evaluación discursiva: la postura epistémica, relacionada con la evaluación del conocimiento, la verdad y el grado de certeza de la información expresada; la postura afectiva, vinculada a la manera en que el hablante o escritor intenta influir en la realidad, en las acciones o en las actitudes de los demás mediante el discurso; y la postura emocional, centrada en la expresión de sentimientos y en la evaluación positiva o negativa de personas, entidades o situaciones. Esta tipología y los criterios asociados se basan en el sistema de anotación desarrollado en el marco del proyecto RACISMMAFF por Carretero y Domínguez [45].

Una distinción fundamental dentro de este sistema es la existente entre uso personal, cuando el hablante se incluye explícitamente como fuente de la evaluación, y uso impersonal, cuando dicha fuente no se menciona de manera directa. Asimismo, el sistema de anotación se articula en torno a tres grandes conjuntos de marcadores: los epistémicos, identificados mediante la etiqueta EP; los efectivos, identificados mediante la etiqueta EF; y los emocionales, identificados mediante la etiqueta EMO.

5.2.1 Epistemic words and expressions

La postura epistémica del hablante o del escritor, entendida como la actitud cognitiva frente a la información transmitida, se define siguiendo los criterios establecidos en el documento de anotación *Epistemic words and expressions*, desarrollado en el marco del proyecto RACISMMAFF [32]. Esta postura no describe los hechos en sí mismos, sino la manera en que estos se presentan en términos de certeza, probabilidad, inferencia, interpretación o desconocimiento.

El sistema distingue entre usos personales, en los que el hablante se identifica explícitamente como fuente del conocimiento, y usos no personales, en los que la fuente se presenta de manera impersonal o generalizada. Las principales categorías de expresiones epistémicas definidas en la documentación son las siguientes:

- **Cognitive Factives <EP, FV>**

Expresiones que presentan una proposición como un hecho verdadero o conocimiento seguro, tanto en usos personales (*I know*) como impersonales (*it is known that*).

- **Cognitive Attitude Predicates <EP, CGA>**

Recursos que expresan opinión, creencia o evaluación subjetiva sin afirmar certeza absoluta, como *I think*, *I believe* o *in my opinion*.

- **Epistemic Modals <EP, EM>**

Elementos que expresan grados de probabilidad o posibilidad, situando la proposición en un continuo que va desde la duda hasta la certeza (*maybe*, *probably*, *might*).

- **Indirect Inferential Evidentiality <EP, IIE>**

Expresiones que indican que la información procede de una inferencia basada en indicios o evidencias indirectas (*it seems*, *apparently*, *it is clear*).

- **Interpretation or Reformulation of Information <EP, IIR>**

Recursos mediante los cuales el hablante interpreta datos o hechos como evidencia que respalda una determinada conclusión (*this suggests, this indicates*).

- **Aphonic or Ignorative Predicates <EP, APH>**

Expresiones que manifiestan falta de conocimiento, duda o ignorancia, tanto de forma personal (*I don't know*) como impersonal (*nobody knows*).

- **Marked Enunciational Positioning <EP, MEP>**

Casos en los que el hablante se posiciona explícitamente en el discurso mediante verbos performativos de decir (*I say, I have to say*).

5.2.2 Effective words and expressions

La descripción de la postura efectiva que se presenta a continuación se basa en los criterios de anotación definidos por Carretero y Domínguez en el documento *Effective words and expressions*, elaborado en el marco del proyecto RACISMMAFF [33].

La postura efectiva se manifiesta mediante la imposición de obligaciones, la expresión de intenciones, la formulación de juicios normativos o la presentación de posibilidades de actuación. Al igual que en el caso epistémico, se distingue entre usos personales y no personales. Las principales categorías definidas son:

- **Directivity <EF, DIR>**

Expresiones que dirigen o incitan a la realización de una acción, incluyendo imperativos y exhortaciones (*look, remember, let's*).

- **Deonticity <EF, DM>**

Recursos que expresan obligación, necesidad, permiso o prohibición, como *must, should, need to* o *it is necessary*.

- **Intentionality <EF, INT>**

Expresiones que reflejan la intención o el compromiso del hablante respecto a una acción futura (*I intend to, we are going to*).

- **Inclination <EF, INC>**

Recursos que expresan deseo, esperanza o voluntad sin implicar obligación firme (*I hope, I wish, I want*).

- **Normativity <EF, NRM>**

Expresiones que evalúan acciones o situaciones en términos de corrección, legitimidad o conveniencia (*it is important, it is urgent*).

- **Potentiality <EF, POT>**

Expresiones que indican posibilidad o capacidad para llevar a cabo una acción, especialmente con fines persuasivos (*can, it is possible to*).

5.2.3 Emotional words and expressions

72 La postura emocional se refiere a la expresión de sentimientos y a la evaluación positiva o negativa de personas, entidades o situaciones. Este tipo de posicionamiento permite no solo expresar emociones, sino también reflexionar sobre sus causas, favoreciendo el desarrollo de una empatía crítica.

77 La caracterización de la postura emocional que se presenta en este apartado se basa en los criterios de anotación definidos en el documento *Emotional words and expressions*, elaborado en el marco del proyecto RACISMMAFF [34]. Dicho sistema toma como referencia la categoría de Actitud de la *Teoría de la Valoración (Appraisal Theory)* propuesta por Martin y White [35], que distingue distintos dominios de evaluación emocional.

- **Afecto:** expresión directa de sentimientos (*I am happy*).
- **Juicio:** evaluaciones éticas del comportamiento humano (*an honest man*).
- **Apreciación:** evaluaciones estéticas o funcionales de objetos o fenómenos (*a fast computer*).

Dentro de este marco teórico, se definen los siguientes subtipos emocionales, utilizados en el sistema de anotación:

- **Security <EMO, SEC>**

Evaluación del grado de seguridad, protección o riesgo asociado a personas o situaciones.

- **Satisfaction <EMO, SAT>**

Expresión de estados de felicidad, satisfacción o, por el contrario, de frustración e insatisfacción.

- **Capacity <EMO, CAP>**

Evaluación de la capacidad, competencia o habilidad de las personas, en relación con sus acciones o comportamientos.

- **Strength <EMO, STR>**

Evaluación de la fortaleza o debilidad, aplicable tanto a personas como a entidades no humanas.

- **Morality <EMO, MOR>**

Evaluaciones emocionales relacionadas con la moralidad, la ética y el comportamiento social, incluyendo juicios sobre veracidad, corrección moral y rasgos de carácter.

- **Quality <EMO, QUA>**

Evaluaciones relacionadas con la calidad, normalidad o valor de personas, objetos, procesos o situaciones, integrando aspectos de apreciación estética, funcional y social.

- **Usefulness <EMO, USE>**

Evaluaciones emocionales vinculadas a la utilidad, la practicidad o la eficacia de acciones, objetos o situaciones, permitiendo expresar si algo se percibe como beneficioso o, por el contrario, como inútil o ineficaz.

5.2.4 Utilidad

Los sistemas de anotación descritos en este apartado se basan en el documento *RACISMMAFF Project: Proposal of Criteria for Including and Excluding Markers for the Annotation of Epistemic and Effective Stance*, elaborado por Marta Carretero y Elena Domínguez Romero.

Las categorías lingüísticas descritas en este apartado no se utilizan únicamente con fines descriptivos, sino que constituyen la base operativa del sistema automático desarrollado en este trabajo. Cada una de las categorías teóricas definidas (EP, EF y EMO, junto con sus subtipos) se traduce directamente en etiquetas del conjunto de datos, que guían el proceso de anotación manual del corpus llevado a cabo por el equipo del proyecto RACISMMAFF. A partir de estas anotaciones, los modelos se entrena para aprender a identificar patrones lingüísticos asociados a cada categoría, de modo que la teoría lingüística se integra explicitamente en el diseño del dataset y en el objetivo de aprendizaje del sistema.

Para su procesamiento automático, resulta fundamental que estos documentos se conviertan a formato .txt, utilizando codificación **UTF-8**, y que se almacenen de manera homogénea en una única carpeta —denominada en este proyecto *txt*—, lo que facilita su tratamiento posterior y la generación de los conjuntos de datos empleados en el entrenamiento de los modelos.

5.3 Desarrollo Datasets

Una vez que todos los textos han sido almacenados y unificados en un mismo formato, se procede a su procesamiento con el objetivo de generar los conjuntos de datos necesarios para el entrenamiento de los modelos. En este proyecto se realizan distintos tipos de procesamiento en función del enfoque de entrenamiento empleado.

5.3.1 Primer sistema de procesamiento

Para el entrenamiento de los modelos *encoder-only* basados en arquitecturas Transformer, los datos se procesan mediante un script denominado `convert_txt.py`. En el punto de partida del proyecto se disponía de un conjunto de scripts heredados de trabajos previos, que habían sido diseñados para cubrir distintas etapas del procesamiento de los datos. No obstante, durante las primeras fases de análisis del sistema se detectaron problemas de funcionamiento derivados de incompatibilidades con versiones antiguas del código, especialmente en el caso del script `convert_txt`. Estas limitaciones

dificultaban la reutilización directa del sistema existente y motivaron la necesidad de replantear el enfoque de procesamiento.

A partir del estudio del flujo de procesamiento heredado y tras evaluar distintas alternativas de reorganización del código, se optó por rediseñar y reimplementar el script desde cero. El nuevo enfoque permitió no solo mejorar el proceso de generación y clasificación de frases, sino también unificar todo el flujo de procesamiento en un único script. En la implementación anterior, el procesamiento se realizaba mediante varios scripts encadenados, cada uno con sus propias entradas y salidas, lo que daba lugar a la creación de múltiples carpetas intermedias. Aunque este enfoque resultaba funcional, la dispersión de los datos a lo largo de distintas etapas hacia más compleja la trazabilidad completa del procesamiento, así como el seguimiento y la depuración de errores. La solución adoptada centraliza todas las funcionalidades en un único punto de control, facilitando la comprensión del proceso completo y la evolución futura del sistema.

En cuanto a las mejoras introducidas en esta nueva versión del procesado, destaca especialmente la revisión del procedimiento de generación de los conjuntos de datos de clasificación. El análisis del sistema previo permitió identificar que uno de los principales problemas estaba relacionado con un modelo RoBERTa EP/EF anteriormente desplegado en Hugging Face, que no proporcionaba puntuaciones de confianza (*scores*), no admitía clasificación multietiqueta (*multilabel*) y forzaba la asignación de una etiqueta incluso cuando la frase analizada no correspondía a la categoría evaluada. Estas limitaciones estaban asociadas a la arquitectura del modelo y al diseño del conjunto de datos utilizado durante su entrenamiento.

En concreto, los datasets procesados presentaban frases incompletas o fuera de su contexto real, no incluían ejemplos multietiqueta (es decir, frases con dos o más etiquetas EP o EF) y carecían de ejemplos sin ninguna etiqueta, ya que todas las frases estaban obligatoriamente etiquetadas. Como consecuencia, el modelo epistémico fue entrenado únicamente con frases etiquetadas como EP, lo que provocaba que cualquier frase introducida, ya fuera EP, EF o ninguna de ellas, fuese clasificada sistemáticamente como EP. Un comportamiento análogo se observó en el entrenamiento de las etiquetas EF.

A partir de este análisis, se redefinió el procedimiento de construcción de los conjuntos de datos, corrigiendo estos problemas mediante la incorporación de frases con mayor contexto discursivo, la inclusión explícita de ejemplos multietiqueta y la adición de frases que no corresponden a la categoría evaluada. De este modo, el modelo aprende a distinguir adecuadamente entre distintos tipos de frases y escenarios posibles, mejorando su capacidad de generalización.

Como resultado de todo este proceso, se desarrolló una nueva versión del script `convert_txt.py`, que constituye la base del procesamiento de datos para los modelos *encoder-only* empleados en el sistema.

A continuación, se presentan capturas del script `convert_txt.py`, que permiten ilustrar algunas de las funciones clave implementadas en este nuevo sistema de procesado.

```
1  import os
2  import json
3  import re
4  import random
5  import argparse
6  import nltk
7  from collections import Counter
8  from nltk.tokenize import sent_tokenize
9
10
11 # Limpiar texto (quita etiquetas <EP, ...>, <EF, ...>, <EMO,...>)
12
13 def clean_text(text: str) -> str:
14     text = re.sub(r"<[^>]+>", "", text)
15     return re.sub(r"\s+", " ", text).strip()
16
17
18 # Extraer etiquetas normalizadas
19
20 def extract_labels(sentence: str, prefix: str):
21     pattern = fr"<s*{prefix}\s*,\s*([>]+)>"
22     matches = re.findall(pattern, sentence)
23
24     labels = []
25     for m in matches:
26         m = m.strip().replace("-", "_")
27
28         m = re.sub(r"^\w{2,3}\s+$", r"\1_P", m)
29         m = re.sub(r"\s+$", "", m)
30
31         parts = [p.strip().upper() for p in re.split(r"\s*,\s*", m) if p.strip()]
32         if not parts:
33             continue
34
35         tipo = parts[0]
36
37         if prefix in {"EP", "EF"} and len(parts) >= 2 and parts[1] == "P":
38             labels.append(f"{prefix}_{tipo}_P")
39             continue
40
41         POLS = {"POS", "NEG", "NEU", "MIX"}
42
43         if len(parts) >= 3 and parts[2] in POLS:
44             labels.append(f"{prefix}_{tipo}_{parts[2]}")
45         elif len(parts) >= 2 and parts[1] in POLS:
46             labels.append(f"{prefix}_{tipo}_{parts[1]}")
47         else:
48             labels.append(f"{prefix}_{tipo}")
49
50     return labels
51
52
53 # SUBMUESTREO DE NEGATIVAS
54
55 def balance_dataset(entries, none_label, max_neg_ratio=1.5, seed=42):
56     random.seed(seed)
57
58     positives = [e for e in entries if none_label not in e["labels"]]
59     negatives = [e for e in entries if none_label in e["labels"]]
60
61     max_negs = int(len(positives) * max_neg_ratio)
62     if len(negatives) > max_negs:
63         negatives = random.sample(negatives, max_negs)
64
65     balanced = positives + negatives
```

Figura 5.3.1.1 Inicio del script *covert_txt.py*

```
66     random.shuffle(balanced)
67     return balanced
68
69
70
71 # CONCATENAR TODOS LOS .txt DE UNA CARPETA
72
73 def concat_txt_folder(folder_path, output_path):
74     files = sorted(f for f in os.listdir(folder_path) if f.lower().endswith(".txt"))
75     if not files:
76         raise ValueError(f"No se han encontrado .txt en la carpeta: {folder_path}")
77
78     with open(output_path, "w", encoding="utf-8") as outfile:
79         for fname in files:
80             with open(os.path.join(folder_path, fname), "r", encoding="utf-8") as infile:
81                 contenido = infile.read().strip()
82                 if contenido:
83                     outfile.write(contenido + "\n\n")
84
85     print(f"✓ input concatenado generado en: {output_path}")
86     return output_path
87
88
89
90 # MAIN
91
92 def main():
93     parser = argparse.ArgumentParser()
94
95     parser.add_argument("--folder", type=str, default=".txt")
96     parser.add_argument("--output_ep", type=str, default="output_ep.json")
97     parser.add_argument("--output_ef", type=str, default="output_ef.json")
98     parser.add_argument("--output_emo", type=str, default="output_emo.json")
99     parser.add_argument("--train_ep", type=str, default="train_ep.json")
100    parser.add_argument("--train_ef", type=str, default="train_ef.json")
101    parser.add_argument("--train_emo", type=str, default="train_emo.json")
102    parser.add_argument("--debug", action="store_true")
103    parser.add_argument("--debug_limit", type=int, default=20)
104
105    args = parser.parse_args()
106
107    nltk.download("punkt", quiet=True)
108    nltk.download("punkt_tab", quiet=True)
109
110    base_dir = os.path.dirname(os.path.abspath(__file__))
111    folder_path = os.path.join(base_dir, args.folder)
112
113    temp_input = os.path.join(base_dir, "_input_concatenado.txt")
114    input_path = concat_txt_folder(folder_path, temp_input)
115
116    with open(input_path, "r", encoding="utf-8") as f:
117        text = f.read()
118
119    sentences = sent_tokenize(text)
120
121    ep_entries, ef_entries, emo_entries = [], [], []
122    train_ep_entries, train_ef_entries, train_emo_entries = [], [], []
123
124    unmatched_emo_counter = Counter()
125    unmatched_examples = []
126
127    for sent in sentences:
128        sent_clean = clean_text(sent)
129
130        ep_labels = extract_labels(sent, "EP")
```

Figura 5.3.1.2 Continuación del script convert_txt.py

```
131     ef_labels = extract_labels(sent, "EF")
132     emo_labels = extract_labels(sent, "EMO")
133
134     if ep_labels:
135         ep_entries.append({"text": sent_clean, "labels": ep_labels})
136     if ef_labels:
137         ef_entries.append({"text": sent_clean, "labels": ef_labels})
138     if emo_labels:
139         emo_entries.append({"text": sent_clean, "labels": emo_labels})
140
141     ep_or_none = ep_labels if ep_labels else ["NO_EP"]
142     ef_or_none = ef_labels if ef_labels else ["NO_EF"]
143     emo_or_none = emo_labels if emo_labels else ["NO_EMO"]
144
145     train_ep_entries.append({"text": sent_clean, "labels": ep_or_none})
146     train_ef_entries.append({"text": sent_clean, "labels": ef_or_none})
147     train_emo_entries.append({"text": sent_clean, "labels": emo_or_none})
148
149     if args.debug:
150         raw_emo_tags = re.findall(r"<\s*EMO\s*,\s*[^\>]+>", sent)
151         if raw_emo_tags and not emo_labels:
152             for t in raw_emo_tags:
153                 unmatched_emo_counter[t] += 1
154             if len(unmatched_examples) < args.debug_limit:
155                 unmatched_examples.append(sent.strip())
156
157     # Balanceo FINAL (aqui es donde decides cuantos NO_* entran de verdad)
158     balanced_train_ep = balance_dataset(train_ep_entries, "NO_EP")
159     balanced_train_ef = balance_dataset(train_ef_entries, "NO_EF")
160     balanced_train_emo = balance_dataset(train_emo_entries, "NO_EMO")
161
162     # Conteo SOLO sobre el dataset balanceado (esto corrige NO_EP/NO_EF/NO_EMO)
163     ep_label_counts_train = Counter(lab for e in balanced_train_ep for lab in e["labels"])
164     ef_label_counts_train = Counter(lab for e in balanced_train_ef for lab in e["labels"])
165     emo_label_counts_train = Counter(lab for e in balanced_train_emo for lab in e["labels"])
166
167     # Guardar datasets
168     for path, data in [
169         (args.output_ep, ep_entries),
170         (args.output_ef, ef_entries),
171         (args.output_emo, emo_entries),
172         (args.train_ep, balanced_train_ep),
173         (args.train_ef, balanced_train_ef),
174         (args.train_emo, balanced_train_emo),
175     ]:
176         with open(path, "w", encoding="utf-8") as f:
177             json.dump(data, f, indent=2, ensure_ascii=False)
178
179     # Guardar conteos TRAIN (balanceados)
180     with open("label_counts_ep_train.json", "w", encoding="utf-8") as f:
181         json.dump(ep_label_counts_train, f, indent=2, ensure_ascii=False)
182     with open("label_counts_ef_train.json", "w", encoding="utf-8") as f:
183         json.dump(ef_label_counts_train, f, indent=2, ensure_ascii=False)
184     with open("label_counts_emo_train.json", "w", encoding="utf-8") as f:
185         json.dump(emo_label_counts_train, f, indent=2, ensure_ascii=False)
186
186     print("✓ Dataset generado correctamente.")
187     print(f"- EP puras: {len(ep_entries)}")
188     print(f"- EF puras: {len(ef_entries)}")
189     print(f"- EMO puras: {len(emo_entries)}")
190
191     print(f"- NO_EP en train final: {ep_label_counts_train.get('NO_EP', 0)}")
192     print(f"- NO_EF en train final: {ef_label_counts_train.get('NO_EF', 0)}")
193     print(f"- NO_EMO en train final: {emo_label_counts_train.get('NO_EMO', 0)}")
```

Figura 5.3.1.3 Continuación del script convert_txt.py

```
194     print(f"- NO_EMO en train final: {emo_label_counts_train.get('NO_EMO', 0)}")  
195  
196     print("Conteos TRAIN balanceados guardados en label_counts_*_train.json")  
197  
198     if args.debug and unmatched_emo_counter:  
199         print("\n[DEBUG EMO] Tags EMO problemáticos:")  
200         for tag, c in unmatched_emo_counter.most_common(20):  
201             print(f"  {c}x {tag}")  
202  
203  
204     if __name__ == "__main__":  
205         main()  
206
```

Figura 5.3.1.4 Final del script convert_txt.py

clean_text:

- Elimina todas las etiquetas del tipo <...> presentes en el texto.
- Normaliza los espacios (convierte múltiples espacios en uno solo).
- Elimina espacios al inicio y al final

extract_labels:

Busca dentro de una frase etiquetas anotadas manualmente del tipo:

- <EP, ...>
- <EF, ...>
- <EMO, ...>

Extrae y normaliza esas etiquetas a un formato uniforme.

balance_dataset:

```
def balance_dataset(entries, none_label, max_neg_ratio=1.5, seed=42):
```

- Aplica submuestreo de ejemplos negativos (NO_*) en un dataset.
- Calcula el máximo permitido de negativos:
- max_negativos = positivos × max_neg_ratio

Esta parte de la inclusión de los NO_* es crucial para nuestros entrenamientos y tener un 1.5 de max_reg_ratio, creo que sería lo estándar ya que si metemos más cantidades de NO_* es muy probable que nuestros modelos de entrenamiento se centren y aprendan más con los NO_* que con los de las etiquetas reales.

concat_txt_folder:

Esta función simplemente lo que nos hace es concatenar todos los txt que se encuentran en la carpeta text en un solo, y así crear un solo dataset.

Main:

Que se encarga de ejecutar todas las funciones anteriores en cierto orden y generándonos los siguientes json.

- output_ep.json, output_ef.json, output_emo.json: que son los dataset pero sin etiquetas NO_*.
- train_ep.json, train_ef.json, train_emo.json : que estos son los dataset con etiquetas NO_* y vamos a utilizar para nuestros entrenamientos.
- label_counts_ep_train.json, label_counts_ef_train.json, label_counts_emo_train.json: estos json simplemente son un conteo de cuantas etiquetas hay de cada tipo para luego en un futuro hacer mejoras, sobre todos en aquellas etiquetas que hay pocas cantidades.

Luego en el main ponemos unos print para tema de dapuración y ver si la ejecución es acorde a lo que queremos, cuyos resultados son:

✓ input concatenado generado en:

c:\Users\palom\OneDrive\Escritorio\TFG\prueba_input_concatenado.txt

✓ Dataset generado correctamente.

- EP puras: 3208
- EF puras: 3583
- EMO puras: 1367
- NO_EP en train final: 4812
- NO_EF en train final: 5374
- NO_EMO en train final: 2050

Conteos TRAIN balanceados guardados en label_counts_*_train.json

Nationality and Borders Bill
The British people have had enough of <EF, NRM> open borders and uncontrolled immigration; enough of <EF, NRM> a failed asylum system.
(–)
Our plan will increase the fairness of our system so that we can <EF, POT, P> better <EF, NRM> protect those who are in need of <EF, NRM>
(–)
For years, people have risked their lives to enter our country, such as those crossing the channel in dangerous boats to claim asylum.
If there were simple and straightforward solutions to many of these challenges-my right hon. Friend the Member for Maidenhead (Mrs.)
The Bill will finally address the issues that over a long period of time, cumulatively, have resulted in the broken system that we
(–)
My right hon. Friend is absolutely right <EF, NRM>. There are many ways in which we can <EF, POT, P> address this problem, and crea
(–) we must <EF, DM, P> recognise <EP, FV, P> that, given the trends we are seeing <EP, IIE, P> in illegal migration, the majority
(–)
Providing greater support to refugees arriving safely will <EP, EM> reduce the incentive to enter the country dangerously and illeg
(–) It is not just illegal <EF, NRM> sea journeys that are lethal. One of my first and saddest tasks as Home Secretary was to respond to
It is illegal to <EF, NRM> arrive in UK waters without permission. Those who bring migrants to the UK and facilitate illegal entry
Le 1 Col 1 Session A UTS Read ROM CRF L1 Plain Text 80 O

Figura 5.3.1.5 Estado inicial del txt

En la figura 5.3.1.5 podemos observar cómo es el estado inicial de nuestro documento txt en formato y-utf-8, para luego pasar a procesarlo y transformarlo,

```
{
  "text": "We do not want to end up with a situation like the South Sudanese in Kenya or Uganda, where children may spend their whole lives in camps.",
  "label": [
    "EP_DM"
  ]
}

{
  "text": "ID cards are practical, popular and essential to securing our borders.",
  "label": [
    "EP_DM",
    "NO_EP"
  ]
}

{
  "text": "Indeed European border guards would bar migrants entry from Greece to a non-EU country, Macedonia.",
  "label": [
    "EP_DM"
  ]
}

{
  "text": "As former head of Interpol Ronald Noble points out, eight EU members are on Interpol's top 20 list of nations reporting stolen or lost passports.",
  "label": [
    "NO_EP"
  ]
}

{
  "text": "If the Government cannot do this-and they certainly have not so far-we are in real danger of passing a Bill that, as it stands, would be counterproductive in respect of illegal immigration and would harm community cohesion.",
  "label": [
    "EP_DM",
    "NO_EP",
    "NO_DM",
    "NO_EP"
  ]
}

{
  "text": "Even more opposed gay marriage, and nearly half thought that it was unacceptable for a gay person to teach their children.",
  "label": [
    "NO_EP"
  ]
}

{
  "text": "We have the tools.",
  "label": [
    "NO_EP"
  ]
}

{
  "text": "What support will be in place for refugees then, if needed?",
  "label": [
    "NO_EP"
  ]
}

{
  "text": "I appreciate that opinions on that may differ, but the purpose is to discourage asylum seekers from travelling to the UK other than via safe and legal routes.",
  "label": [
    "EP_DM",
    "NO_EP"
  ]
}

{
  "text": "There is talk of Ukrainians \"not really wanting to be permanently settled here\" and of the need to submit them to official security clearance and biometric data checking.",
  "label": [
    "NO_EP"
  ]
}
```

Figura 5.3.1.6 Resultado del tain_ep.json

En la figura 5.3.1.6 podemos observar la estructura el resultado final de los datos de etiquetado EP para su futuro entrenamiento.

```

20041 "text": "This is the crucial point of this Bill.",  
20042 "label": [  
20043 "IF_SOM"  
20044 ]  
20045 }  
20046  
20047 "text": "Now might be identity cards, though I would still argue that they should be voluntary, offered as a cheap version of a passport.",  
20048 "label": [  
20049 "IF_POT",  
20050 "IF_DM"  
20051 ]  
20052 }  
20053  
20054 "text": "The scandal revealed practices which should shame those responsible.",  
20055 "label": [  
20056 "IF_POT"  
20057 ]  
20058 }  
20059  
20060 "text": "Of course the British military should be helping to protect our coastline bringing in the Navy to support Border Force officials is a no-brainer.",  
20061 "label": [  
20062 "IF_DM"  
20063 ]  
20064 }  
20065  
20066 "text": "The commission urged us to look at the facts before deciding.",  
20067 "label": [  
20068 "IF_POT"  
20069 ]  
20070 }  
20071  
20072 "text": "A day or two later I met the same couple on the Glasgow train.",  
20073 "label": [  
20074 "NO_ITP"  
20075 ]  
20076 }  
20077  
20078 "text": "If it is not Syria or Afghanistan, it will be somewhere else.",  
20079 "label": [  
20080 "NO_ITP"  
20081 ]  
20082 }  
20083  
20084 "text": "Its economy is expected to contract by 30 per cent this year.",  
20085 "label": [  
20086 "NO_ITP"  
20087 ]  
20088 }  
20089  
20090 "text": "The Foreign Secretary has just said that it might be in Itille, nearly 75 miles from Calais.",  
20091 "label": [  
20092 "NO_ITP"  
20093 ]  
20094 }  
20095  
20096 "text": "The world's focus must be on finding political solutions to the cycles of violence that drive civilians from their homes, and on breaking the culture of impunity that has come to characterise brutal conflicts such as those in Syria and South Sudan.",  
20097

```

Figura 5.3.1.7 Resultado del tain_ef.json

En la figura 5.3.1.7 podemos observar la estructura el resultado final de los datos de etiquetado EF para su futuro entrenamiento.

```

511 "text": "A year later, the high court said the facilities there did not meet "minimum standards".",  
512 "label": [  
513 "NO_DM_NG"  
514 ]  
515 }  
516  
517 "text": "Teachers described the refugee girls as exceptionally talented.",  
518 "label": [  
519 "NO_DM_PSG"  
520 ]  
521 }  
522  
523 "text": "115L.",  
524 "label": [  
525 "NO_DM"  
526 ]  
527 }  
528  
529 "text": "The lack of translation services felt insensitive to new arrivals.",  
530 "label": [  
531 "NO_DM_NG"  
532 ]  
533 }  
534  
535 "text": "Why the caption?",  
536 "label": [  
537 "NO_DM"  
538 ]  
539 }  
540  
541 "text": "Workers felt resentful of the unfair treatment.",  
542 "label": [  
543 "NO_DM_NG"  
544 ]  
545 }  
546  
547 "text": "I would much rather I were teaching English in Bosnia after a life of peace, as I had imagined and hoped, than speaking here, after an experience of war.",  
548 "label": [  
549 "NO_DM"  
550 ]  
551 }  
552  
553 "text": "The council's handling of the complaint seemed correct , but its spirit remained open to interpretation.",  
554 "label": [  
555 "NO_DM_NG"  
556 ]  
557 }  
558  
559 "text": "In recent times, the main commodity smuggled has been people.",  
560 "label": [  
561 "NO_DM"  
562 ]  
563 }  
564  
565 "text": "If it is not Syria or Afghanistan, it will be somewhere else.",  
566 "label": [  
567 "NO_ITP"  
568 ]  
569 }

```

Figura 5.3.1.8 Resultado del tain_emo.json

En la figura 5.3.1.8 podemos observar la estructura el resultado final de los datos de etiquetado EMO para su futuro entrenamiento.

5.3.2 Segundo sistema de procesamiento

91 A continuación, se describe el segundo sistema de procesamiento de datos, destinado al entrenamiento de modelos *encoder-decoder* basados en arquitecturas Transformer. En una primera aproximación, el procesamiento se apoyaba en dos scripts principales: process_txt.py, encargado de la extracción y preparación de las frases a partir de los textos originales, y change_labels.py, responsable de la armonización y transformación de las etiquetas al formato requerido por el modelo.

Al igual que ocurrió con el script convert_txt.py, los scripts proporcionados inicialmente no correspondían a versiones plenamente funcionales o actualizadas, lo que hizo necesaria su recodificación. Durante este proceso se introdujeron mejoras significativas, entre las que destacan la ampliación del contexto de las frases extraídas y la generación de salidas que incorporan etiquetas no contempladas en las versiones originales de los scripts, con el objetivo de mejorar la calidad y cobertura del conjunto de datos generado.

5 No obstante, el análisis del flujo completo de procesamiento mostró que la separación en dos scripts consecutivos generaba una dependencia estricta entre ambas etapas y la creación de archivos intermedios, lo que añadía complejidad al proceso sin aportar beneficios claros en términos de flexibilidad o reutilización. Por este motivo, se optó por unificar ambas funcionalidades en un único script, denominado process_txt.py, que integra tanto el procesado de los textos como la armonización de las etiquetas en una sola ejecución.

Para el entrenamiento de los modelos *encoder-decoder* se construyó un conjunto de datos diferente al utilizado en los modelos *encoder-only*. En este caso, el dataset se estructura en pares *input-output*, donde el *input* corresponde a la frase original sin etiquetas y el *output* a la misma frase anotada con la etiqueta correspondiente. Este formato resulta especialmente adecuado para modelos *encoder-decoder*, cuyo objetivo es generar directamente la secuencia anotada a partir del texto de entrada, en lugar de asignar etiquetas de forma independiente.

Como resultado de este rediseño, el nuevo sistema de procesado permite generar conjuntos de datos coherentes y consistentes en una única ejecución, facilitando la trazabilidad del proceso, su mantenimiento y la reproducibilidad de los experimentos realizados con modelos generativos.

```
❸ process_txt.py > ...
1  import os
2  import re
3  import json
4  import nltk
5  from nltk.tokenize import sent_tokenize
6
7
8  TXT_FOLDER = "txt"
9
10 # Si quieres guardar también los JSON intermedios (data_ep.json / data_ef.json)
11 SAVE_INTERMEDIATE = False
12
13 DATA_EP = "data_ep.json"
14 DATA_EF = "data_ef.json"
15
16 DATA_READY_EP = "data_ready_ep.json"
17 DATA_READY_EF = "data_ready_ef.json"
18
19
20 # REGEX
21 TAG_EP_PATTERN = re.compile(r"<EP\s*,\s*[^\>]+>")
22 TAG_EF_PATTERN = re.compile(r"<EF\s*,\s*[^\>]+>")
23 REMOVE_CURLY_PATTERN = re.compile(r"\{\[^{\}]*\}")
24 REMOVE_TAGS = re.compile(r"<[^>]+>")
25
26
27 def ensure_sentence_end(s: str) -> str:
28     s = s.rstrip()
29     if not s:
30         return s
31     if s.endswith((".", "!", "?")):
32         return s
33     if s.endswith(";", ":", ",") or s.endswith("..."):
34         return s[:-1].rstrip() + "."
35     if s[-1] in ')}'":
36         if len(s) > 1 and s[-2] in ".!?"':
37             return s
38         if len(s) > 1 and s[-2] in ";,:":
39             return s[:-2].rstrip() + s[-1]
40         return s[:-1].rstrip() + "." + s[-1]
41     return s + "."
42
43 def clean_text(text: str) -> str:
44     text = REMOVE_CURLY_PATTERN.sub("", text)
45     text = REMOVE_TAGS.sub("", text)
46     text = re.sub(r"\s+", " ", text).strip()
47     return text
48
49 def process_text_file(filepath: str):
50     with open(filepath, "r", encoding="utf-8") as f:
51         content = f.read()
```

Figura 5.3.2.1 Script process_txt.py

```
52     # mismo delimitador de frases
53     sentences = sent_tokenize(content)
54
55     ep_items = []
56     ef_items = []
57
58     for sentence in sentences:
59         has_ep = bool(TAG_EP_PATTERN.search(sentence))
60         has_ef = bool(TAG_EF_PATTERN.search(sentence))
61         if not (has_ep or has_ef):
62             continue
63
64         output_full = ensure_sentence_end(sentence.strip())
65         input_full = ensure_sentence_end(clean_text(output_full))
66
67         item = {
68             "input_text": input_full,
69             "output_text": output_full
70         }
71
72         if has_ep:
73             ep_items.append(item)
74         if has_ef:
75             ef_items.append(item)
76
77     return ep_items, ef_items
78
79
80
81     # normalización de tags
82     # (convierte <EP, a, b> -> EP_a_b y elimina tags del otro tipo)
83
84     def normalize_items(main_label: str, items: list[dict]) -> list[dict]:
85         other_label = "EF" if main_label == "EP" else "EP"
86
87         main_tag_pattern = re.compile(
88             rf"<\s*{main_label}\s*,\s*(^,\s*)+\s*(?:,\s*(^>\s*)+)?\s*>"
89         )
90         other_tag_pattern = re.compile(
91             rf"<\s*{other_label}\s*,\s*[^>]+>"
92         )
93
94         def replace_main_tags(text: str) -> str:
95             def repl(match):
96                 part1 = match.group(1).strip()
97                 part2 = match.group(2).strip() if match.group(2) else None
98                 return f"{main_label}_{part1}_{part2}" if part2 else f"{main_label}_{part1}"
99             return main_tag_pattern.sub(repl, text)
```

Figura 5.3.2.2 Script process_txt.py

```
99     |     return main_tag_pattern.sub(repl, text)
100
101    def remove_other_tags(text: str) -> str:
102        |     return other_tag_pattern.sub("", text)
103
104    updated = []
105    for item in items:
106        |     text = item.get("output_text", "")
107
108        if main_tag_pattern.search(text) or other_tag_pattern.search(text):
109            new_text = replace_main_tags(text)
110            new_text = remove_other_tags(new_text)
111            new_text = re.sub(r"\s+", " ", new_text).strip()
112
113            item = dict(item) # copia por seguridad
114            item["output_text"] = new_text
115            updated.append(item)
116
117    return updated
118
119 def main():
120     nltk.download("punkt", quiet=True)
121
122     all_ep = []
123     all_ef = []
124
125     for filename in os.listdir(TXT_FOLDER):
126         if filename.endswith(".txt"):
127             filepath = os.path.join(TXT_FOLDER, filename)
128             print(f"Procesando: {filepath}")
129             ep, ef = process_text_file(filepath)
130             all_ep.extend(ep)
131             all_ef.extend(ef)
132
133     print(f"\nExtraídos: EP={len(all_ep)} | EF={len(all_ef)}")
134
135     # (Opcional) guardar intermedios
136     if SAVE_INTERMEDIATE:
137         with open(DATA_EP, "w", encoding="utf-8") as f:
138             json.dump(all_ep, f, indent=4, ensure_ascii=False)
139         with open(DATA_EF, "w", encoding="utf-8") as f:
140             json.dump(all_ef, f, indent=4, ensure_ascii=False)
141         print(f"Intermedios guardados: {DATA_EP}, {DATA_EF}")
```

Figura 5.3.2.3 Script process_txt.py

```
143     # Normalizar tags y guardar outputs finales
144     ready_ep = normalize_items("EP", all_ep)
145     ready_ef = normalize_items("EF", all_ef)
146
147     with open(DATA_READY_EP, "w", encoding="utf-8") as f:
148         json.dump(ready_ep, f, indent=4, ensure_ascii=False)
149
150     with open(DATA_READY_EF, "w", encoding="utf-8") as f:
151         json.dump(ready_ef, f, indent=4, ensure_ascii=False)
152
153     print(f"\n✓ {DATA_READY_EP} generado con {len(ready_ep)} ítems (procesando EP)")
154     print(f"✓ {DATA_READY_EF} generado con {len(ready_ef)} ítems (procesando EF)")
155
156     if __name__ == "__main__":
157         main()
158
```

Figura 5.3.2.4 Script process_txt.py

def ensure_sentence_end()

Su misión es garantizar que cada frase termina bien.
Si acaba en . ! ? no toca nada.
Si acaba en ; : , ... lo cambia por .
Si acaba en un cierre)}"" intenta colocar el punto antes del cierre si falta.
Texto original:
"Hola mundo"
Texto resultado:
"Hola mundo."

def clean_text(text):

Produce una versión limpia del texto:

{...} , <EP,...> / <EF,...>

def process_text_file(filepath):

Lee un archivo .txt.
Lo divide en frases.
Selecciona solo las frases que tienen <EP,...> o <EF,...>
Crea pares input → output para entrenamiento
Devuelve dos listas: una para EP y otra para EF

normalize_items(main_label: str, items: list[dict])

Hace la normalización final de etiquetas (equivale al antiguo change_labels pero integrado).
Convierte las etiquetas del tipo principal a formato estándar:
<EP, CGA, P> → EP_CGA_P
<EF, X> → EF_X

main():

Inicializa lo necesario (NLTK), crucial para nuestro entrenamiento.
Recorre todos los .txt de la carpeta.
Convierte cada archivo en ejemplos EP / EF.
Une todos los ejemplos.
Guarda dos datasets JSON finales, data_ep.json y data_ef.json.

Con estos dos json generados, los pasamos ahora por el script change_labels, simplemente para limpiar un poco el estilo de las etiquetas y que sigan un estándar de pasar de <EP, CGA, P> a EP_CGA_P.

Dáandonos como resultado los dataset data_ready_ep.json y data_ready_ef.json como en el formato mostrado en la foto de abajo, listos para entrenar.

```

10  {
11    "input_text": "Friend the Member for Maidenhead (Mrs May) has touched on this issues such as illegal migration to the UK would have been resolved by now, but illegal entry to the UK and the subsequent claims of asylum have become complex because of the nature of the issue.",  

12    "output_text": "Friend the Member for Maidenhead (Mrs May) has touched on this issues such as illegal migration to the UK would EP_DM have been resolved by now, but illegal entry to the UK and the subsequent claims of asylum have become complex because of the nature of the issue.",  

13  },  

14  {
15    "input_text": "But I am absolutely clear that no one should seek to put their life, or the lives of their family, in the hands of criminals to enter the UK illegally, and I would like to think that that is an important point that this House can unite on.",  

16    "output_text": "But I am absolutely clear that EP_CQA_P no one should seek to put their life, or the lives of their family, in the hands of criminals to enter the UK illegally, and I would like to think that that is an important point that this House can unite on.",  

17  },  

18  {
19    "input_text": "(...) we must recognise that, given the trends we are seeing in illegal migration, the majority of people entering the UK illegally are travelling through safe countries across the EU where they could claim asylum.",  

20    "output_text": "(...) we must recognise EP_IV_P that, given the trends we are seeing EP_ITF_P in illegal migration, the majority of people entering the UK illegally are travelling through safe countries across the EU where they could claim asylum.",  

21  },  

22  {
23    "input_text": "Indeed , the figures bear this out.",  

24    "output_text": "Indeed EP_CQA , the figures bear this out.",  

25  },  

26  {
27    "input_text": "That is why--If I may just finish my remarks before I go back to my speech .",  

28    "output_text": "That is why--If I may just finish my remarks before I go back to my speech EP_NECP ."  

29  },  

30  {
31    "input_text": " (...) Providing greater support to refugees arriving safely will reduce the incentive to enter the country dangerously and illegally , because when the British people object to illegal entry, they are right to make the case as to it being absolute .",  

32    "output_text": " (...) Providing greater support to refugees arriving safely will EP_EM reduce the incentive to enter the country dangerously and illegally , because when the British people object to illegal entry, they are right to make the case as to it being absolute .",  

33  },  

34  {
35    "input_text": "Those who claim that it is heartless to stop these illegal crossings have it all the wrong way round , because it would be heartless and immoral to let them continue to do so through these dangerous and perilous journeys .",  

36    "output_text": "Those who claim that it is heartless to stop these illegal crossings have it all the wrong way round , because it would EP_EM be heartless and immoral to let them continue to do so through these dangerous and perilous journeys ."  

37  },  

38  {
39    "input_text": "One of my first and saddest tasks as Home Secretary was to respond to the devastating and, really , preventable deaths of 39 Vietnamese people in a trailer found in Essex .",  

40    "output_text": "One of my first and saddest tasks as Home Secretary was to respond to the devastating and, really EP_FV , preventable deaths of 39 Vietnamese people in a trailer found in Essex ."  

41  },  

42  {
43    "input_text": "For me , that is the most important provision in the Bill .",  

44    "output_text": "For me EP_GA_P , that is the most important provision in the Bill ."  

45  },

```

Figura 5.3.2.5 Json resultado después de ejecutar change_labels.py

En la figura 5.3.2.5 podemos observar la estructura el resultado final de los datos para pasar al entrenamiento encoder-decoder.

5.3.3 Tercer sistema de procesamiento

El desarrollo del tercer sistema de procesamiento de datos estuvo motivado por las particularidades de la categoría de posicionamiento emocional (EMO) y por las limitaciones detectadas al aplicar, de forma directa, los mismos métodos de procesado y entrenamiento utilizados previamente para las categorías epistémica (EP) y efectiva (EF).

La documentación y los ejemplos de anotación correspondientes a la categoría EMO fueron incorporados al proyecto en una fase más avanzada de su desarrollo. En una primera aproximación, se intentó aplicar el mismo enfoque empleado para EP y EF, basado en modelos *encoder-only* y *encoder-decoder* para la clasificación a nivel de frase. Sin embargo, durante las fases iniciales de experimentación se observó que los resultados obtenidos no alcanzaban un rendimiento satisfactorio.

Esta limitación estaba directamente relacionada con la menor cantidad de ejemplos disponibles para la categoría emocional, así como con la distribución poco equilibrada de las etiquetas, en la que predominaban los ejemplos asociados a una única *label* frente a frases *multilabel*. Como consecuencia, los modelos entrenados bajo este enfoque mostraban dificultades para generalizar y para identificar de forma consistente las expresiones emocionales en el texto.

Ante esta situación, se optó por mantener el uso de modelos *encoder-only*, pero modificando el enfoque de procesado y entrenamiento. En lugar de realizar decisiones discretas a nivel de frase, se adoptó un esquema de etiquetado BIO (*Begin–Inside–Outside*) a nivel de token, que permite capturar de manera más fina la información contextual distribuida a lo largo del texto. Este enfoque resulta especialmente adecuado en escenarios con datos limitados, ya que incrementa el número de instancias de aprendizaje y reduce la fragmentación del análisis.

Para implementar este sistema, se desarrollaron tres scripts específicos: process_bio.py, labels_bio.py y clean_bio.py. Los dos primeros mantienen una estructura general similar a la de los scripts utilizados en los sistemas de procesado anteriores, aunque incorporan modificaciones sustanciales en el formato de las anotaciones para adaptarlas al esquema BIO. El script clean_bio.py se encarga de la normalización y limpieza de las anotaciones emocionales, corrigiendo inconsistencias relacionadas con espacios, saltos de línea y formatos irregulares que impedían una alineación fiable entre tokens y etiquetas.

Como resultado de este tercer sistema de procesado, se obtuvo un conjunto de datos específicamente adaptado al entrenamiento de modelos encoder-only en tareas de *sequence labeling* con etiquetado BIO, permitiendo mejorar la estabilidad y coherencia del análisis emocional sin introducir cambios innecesarios en la arquitectura del modelo.

```
{  
    "input_text": "The council presented a balanced, harmonious message of inclusion.",  
    "output_text": "The council presented a balanced, harmonious <>EMO_QUA_POS></> message of inclusion."  
},  
{  
    "input_text": "The pundit used a dull stereotype to describe refugee families.",  
    "output_text": "The pundit used a dull <>EMO_QUA_NEG></> stereotype to describe refugee families."  
},  
[  
    {  
        "input_text": "His portrayal of migrants was disgustingly grotesque .",|  
        "output_text": "His portrayal of migrants was disgustingly grotesque <>EMO_QUA_NEG></>."  
    },  
    {  
        "input_text": "The editorial adopted a boring and xenophobic line.",  
        "output_text": "The editorial adopted a boring <>EMO_QUA_NEG></> and xenophobic line."  
    },  
    {  
        "input_text": "The speech unfolded with an irregular incoherence that revealed prejudice.",  
        "output_text": "The speech unfolded with an irregular <>EMO_QUA_NEG></> incoherence that revealed prejudice."  
    },  
    {  
        "input_text": "The column was condemned for its distastefully exaggerated depiction of asylum seekers.",  
        "output_text": "The column was condemned for its distastefully exaggerated <>EMO_QUA_NEG></> depiction of asylum seekers."  
    },  
]
```

Figura 5.3.3.1 Json resultado después de ejecutar todo el proceso de limpieza para EMO

En la figura 5.3.3.1 podemos observar la estructura el resultado final de los datos para pasar al entrenamiento BIO.

En comparación con los archivos JSON generados mediante el script process_txt.py, la principal diferencia reside en la estructura del etiquetado, diseñada para facilitar la detección de las expresiones emocionales por parte de los modelos entrenados bajo el enfoque BIO. Tras aplicar estas correcciones y normalizaciones, los datasets resultantes quedan preparados para el entrenamiento de los modelos emocionales, permitiendo obtener resultados coherentes y alineados con los objetivos del sistema.

5.4 Entrenamientos

Una vez que los datos fueron procesados y estructurados en sus correspondientes archivos JSON, se procedió a la fase de entrenamiento de los modelos de procesamiento del lenguaje natural. Para ello, se emplearon dos enfoques de entrenamiento basados en arquitecturas Transformer: modelos *encoder-only* y modelos *encoder-decoder*. Cada uno de estos enfoques responde

a objetivos distintos dentro del sistema, por lo que se optó por una combinación de ambos con el fin de aprovechar sus respectivas fortalezas.

Esta decisión no fue arbitraria, sino que se basó en el diseño adoptado en trabajos previos desarrollados en el marco de la plataforma DeStance, en los que la combinación de ambos tipos de modelos demostró un funcionamiento estable y eficaz. Asimismo, esta estrategia resulta coherente con el estado del arte en procesamiento del lenguaje natural, donde se ha demostrado que los modelos *encoder-only* son especialmente adecuados para tareas discriminativas y de etiquetado, mientras que los modelos *encoder-decoder* presentan un mejor comportamiento en tareas generativas y de transformación de secuencias.

Cabe destacar que los procesos de entrenamiento asociados a estos modelos presentan una elevada demanda computacional, especialmente en términos de tiempo de ejecución y uso de recursos, incluso en configuraciones de entrenamiento básicas. Por este motivo, el uso de aceleración por hardware resultó necesario para garantizar la viabilidad del proceso experimental.

Como solución a estas limitaciones, todos los entrenamientos se llevaron a cabo utilizando la plataforma Google Colab, que proporciona acceso a recursos computacionales en la nube, incluyendo CPUs, GPUs y TPUs. En el contexto de este trabajo, el uso de GPUs permitió reducir significativamente los tiempos de entrenamiento y facilitar la ejecución de múltiples experimentos sin necesidad de disponer de infraestructura local especializada.

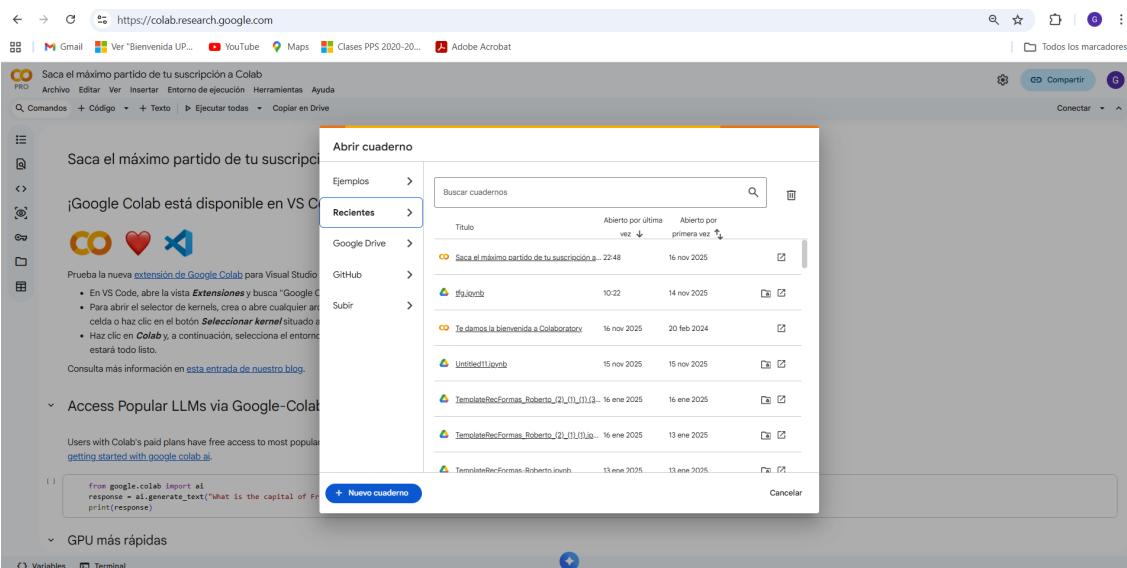


Figura 5.4 .1 Plataforma Google Colab

En la figura 5.4.1 se nos muestra el estado de inicio nada más cliquear en el Google coolab, en este caso ya tengo mi cuaderno creado pero lo suyo seria crearse un cuaderno nuevo.

```

import os
import json
import numpy as np
import torch

from datasets import Dataset

from transformers import (
    AutoTokenizer,
    AutoModelForSeq2SeqLM,
    DataCollatorForSeq2Seq,
    Seq2SeqTrainingArguments,
    Seq2SeqTrainer,
    set_seed,
)

set_seed(42)

# Cargar data_ready_emo.json
DATA_PATH = "data_ready_ef.json"
MODEL_CHECKPOINT = "google/flan-t5-base"
SAVE_DIR = "t5_emo_final"

with open(DATA_PATH, "r", encoding="utf-8") as f:
    raw = json.load(f)

inputs = ["classify emoji:" + ex["input_text"] for ex in raw]
targets = [ex["output_text"] for ex in raw]

print(f"Total ejemplos: {len(inputs)}")

# Crear Dataset HF y split train / valid
dataset = Dataset.from_dict({
    "input_text": inputs,
    "output_text": targets
})

```

Figura 5.4 .2 Plataforma Google Colab

En la figura 5.4.2 muestra ya como es un cuaderno ynb por dentro en el que se puede observar la carpeta sample_data que te viene de serie y es donde vamos a cargar todos los datasets que hemos creado antes. Ya que ahí es donde nuestro script va a buscar y cargar los json con DATA_PATH = "data_ready_ef.json" en este ejemplo.

De base Google colab te va a otorgar un entorno de ejecución base, de CPU, para poder cambiar a otro entorno más rápido hay que ir entorno de ejecución -> cambiar entorno de ejecución y ahí te saldrá todos los entornos disponibles.

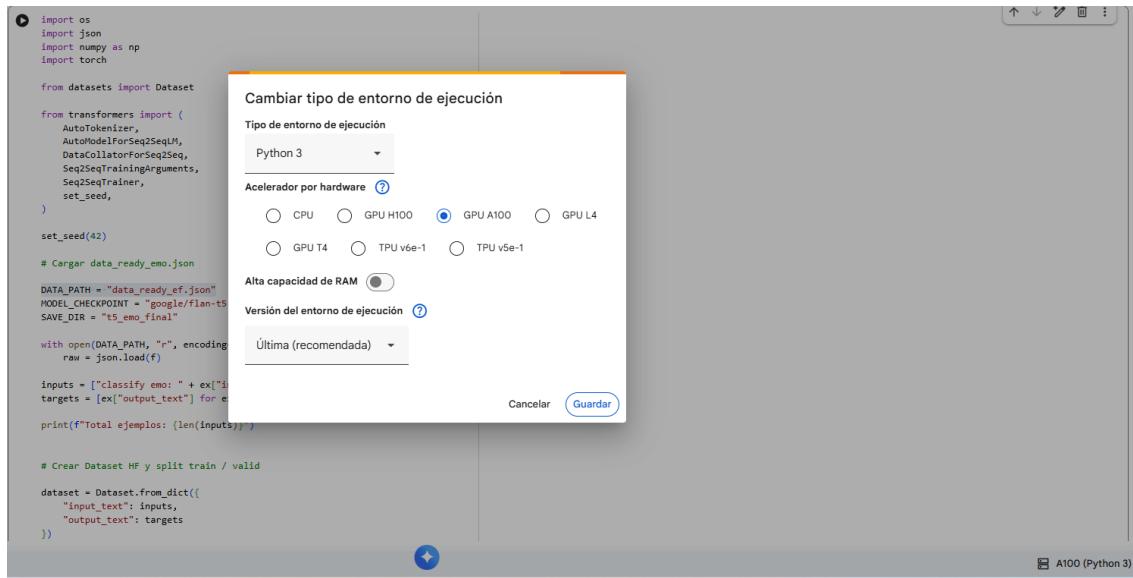


Figura 5.4.3 Maquinas que otorga Google Colab

En la Figura 5.4.3 se muestran los distintos entornos de ejecución disponibles en Google Colab. La plataforma impone, en su versión estándar, ciertas limitaciones en el número de ejecuciones diarias y en el acceso a recursos computacionales avanzados. No obstante, el uso de entornos con GPU permitió acelerar significativamente los procesos de entrenamiento.

5.4.1 Encoder-only Transformer

Los modelos Transformer de tipo encoder-only se emplean principalmente en tareas de comprensión y análisis de texto, ya que su arquitectura está diseñada para generar representaciones semánticas profundas de la entrada mediante mecanismos de atención bidireccional. En este trabajo, este tipo de modelos se utiliza para tareas de clasificación, detección y análisis contextual de patrones lingüísticos, permitiendo identificar información relevante sin necesidad de generar texto de salida.

En versiones previas del sistema se había determinado que RoBERTa-base era el modelo con mejor rendimiento. Sin embargo, dado que en este trabajo se ha llevado a cabo una modificación sustancial del conjunto de datos —incluyendo un rediseño del esquema de anotación y un enfoque multilabel—, ha sido necesario realizar una nueva comparativa desde cero entre distintos modelos encoder-only. Todos los modelos evaluados se han entrenado bajo las mismas condiciones experimentales y con los mismos hiperparámetros básicos estándar, con el objetivo de garantizar la comparabilidad de los resultados y seleccionar posteriormente el modelo con mejor comportamiento para su optimización.

Los modelos encoder-only evaluados en esta fase han sido los siguientes:

- BERT [28]
- DeBERTa [36]
- ELECTRA [37]
- RoBERTa [38]

Todos los modelos se han entrenado y evaluado de forma independiente para cada una de las dimensiones del análisis: postura epistémica (EP), postura efectiva (EF) y postura emocional (EMO).

Antes de presentar los resultados obtenidos y la selección final del modelo, se describen a continuación las métricas empleadas para la evaluación del rendimiento. Estas métricas se calculan utilizando las funciones proporcionadas por la librería scikit-learn [39] y permiten analizar el comportamiento de los modelos desde distintas perspectivas [40].

F1 Macro:

Se trata de la métrica principal utilizada en este trabajo, ya que evalúa el rendimiento del modelo de forma independiente para cada etiqueta y calcula la media sin ponderación. De este modo, todas las clases tienen el mismo peso, lo que resulta especialmente adecuado en escenarios con desbalanceo entre etiquetas.

Recall Macro:

Mide la capacidad del modelo para identificar correctamente las etiquetas reales presentes en el conjunto de datos, es decir, cuántas instancias relevantes son detectadas.

Precision Macro:

Evaluá la proporción de predicciones positivas que son correctas, penalizando la aparición de falsas detecciones o ruido en las predicciones del modelo.

F1 Micro:

Refleja el rendimiento global del modelo agregando todas las decisiones de clasificación. Esta métrica tiende a verse influida por las clases mayoritarias.

Validation Loss:

Mide el error del modelo sobre el conjunto de validación y resulta útil para detectar posibles fenómenos de sobreajuste durante el entrenamiento.

Training Loss:

Indica el error del modelo sobre los datos de entrenamiento y permite observar si el proceso de aprendizaje progresó de forma adecuada.

Epoch:

Representa una pasada completa del modelo por todo el conjunto de entrenamiento.

5.4.1.1 Comparación de entrenamientos Encoder-only Transformer (EP)

En esta sección se analizan los resultados obtenidos tras el entrenamiento de distintos modelos *encoder-only* basados en transformadores, con el objetivo de seleccionar la arquitectura más adecuada para su integración en la plataforma DeStance. Para ello, se evaluaron cuatro modelos en su versión base, comparando su rendimiento a lo largo de varias épocas de entrenamiento.

Dado el posible desbalanceo entre clases, se ha priorizado el uso del **F1 Macro** como métrica principal de evaluación.

A continuación, se presentan los resultados obtenidos por el modelo **bert-base-uncased** durante el proceso de entrenamiento.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.132800 | 0.125917 | 0.226825 | 0.752131 | 0.346938 | 0.201335 |
| 2 | 0.096100 | 0.095529 | 0.419122 | 0.816936 | 0.532375 | 0.376375 |
| 3 | 0.064600 | 0.086524 | 0.581723 | 0.837846 | 0.637024 | 0.551042 |
| 4 | 0.028600 | 0.092775 | 0.606312 | 0.844129 | 0.647580 | 0.577161 |
| 5 | 0.022600 | 0.095338 | 0.624302 | 0.841919 | 0.648719 | 0.606451 |
| 6 | 0.020000 | 0.105720 | 0.627284 | 0.844198 | 0.645794 | 0.611709 |
| 7 | 0.013700 | 0.107364 | 0.623731 | 0.841676 | 0.630198 | 0.619352 |
| 8 | 0.012600 | 0.106664 | 0.624948 | 0.843613 | 0.635509 | 0.615613 |

Figura 5.4.1.1.1 Resultados bert-base-uncased EP

En la Figura 5.4.1.1.1 se muestran los resultados obtenidos por el modelo bert-base-uncased durante el entrenamiento. En ella se observa una disminución

progresiva de la training loss, mientras que la validation loss comienza a estabilizarse a partir de la época 6.

El análisis del F1 Macro indica que el mejor rendimiento se alcanza precisamente en la época 6, con un valor de 0.62784, no observándose mejoras significativas en épocas posteriores. Por este motivo, se selecciona dicha época como la mejor configuración para este modelo.

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo **microsoft/deberta-v3-base**.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.143700 | 0.125649 | 0.142713 | 0.743907 | 0.142535 | 0.143250 |
| 2 | 0.105800 | 0.099360 | 0.330784 | 0.810611 | 0.425455 | 0.309712 |
| 3 | 0.084200 | 0.091154 | 0.400634 | 0.820513 | 0.457743 | 0.377817 |
| 4 | 0.045700 | 0.099608 | 0.446175 | 0.842486 | 0.485328 | 0.415223 |
| 5 | 0.040400 | 0.104821 | 0.468977 | 0.836046 | 0.523838 | 0.451456 |
| 6 | 0.035600 | 0.111782 | 0.541568 | 0.846724 | 0.655660 | 0.506515 |
| 7 | 0.029000 | 0.112753 | 0.581374 | 0.848272 | 0.607791 | 0.564626 |
| 8 | 0.023600 | 0.113670 | 0.597432 | 0.850426 | 0.626637 | 0.577715 |

Figura 5.4.1.1.2 Resultados microsoft/deberta-v3-base EP

En la Figura 5.4.1.1.2 se muestra la evolución de las métricas del modelo microsoft/deberta-v3-base a lo largo de las distintas épocas de entrenamiento. La training loss disminuye de forma progresiva, mientras que la validation loss presenta ligeras fluctuaciones sin un incremento pronunciado, lo que indica un entrenamiento estable.

El análisis del F1 Macro muestra que el mejor rendimiento se alcanza en la época 8, con un valor de 0.597432. Dado que en épocas anteriores no se obtiene un valor superior para esta métrica, se selecciona dicha época como la mejor configuración del modelo para su posterior comparación.

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo **electra-base-discriminator**.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.163000 | 0.141448 | 0.132827 | 0.692815 | 0.138785 | 0.127932 |
| 2 | 0.128000 | 0.117151 | 0.239524 | 0.756352 | 0.252478 | 0.237605 |
| 3 | 0.096700 | 0.101298 | 0.367654 | 0.811180 | 0.409402 | 0.340581 |
| 4 | 0.054400 | 0.105176 | 0.385750 | 0.817518 | 0.488774 | 0.362686 |
| 5 | 0.049800 | 0.102844 | 0.470275 | 0.830555 | 0.544304 | 0.440859 |
| 6 | 0.040400 | 0.108825 | 0.525519 | 0.838384 | 0.578798 | 0.501679 |
| 7 | 0.040500 | 0.111932 | 0.586007 | 0.835548 | 0.645798 | 0.552356 |
| 8 | 0.030700 | 0.111494 | 0.589236 | 0.839375 | 0.620725 | 0.568796 |

Figura 5.4.1.1.3 Resultados electra-base-discriminator EP

En la Figura 5.4.1.1.3 se muestra la evolución de las métricas del modelo electra-base-discriminator a lo largo de las distintas épocas de entrenamiento. Se observa una disminución progresiva de la training loss, mientras que la validation loss presenta ligeras variaciones, manteniéndose relativamente estable en las últimas épocas.

El análisis del F1 Macro indica que el mejor rendimiento se alcanza en la época 8, con un valor de 0.589236.

2

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo **RoBERTa -base**.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.127100 | 0.114416 | 0.304180 | 0.802119 | 0.342985 | 0.282233 |
| 2 | 0.090700 | 0.089792 | 0.514897 | 0.832587 | 0.587403 | 0.486747 |
| 3 | 0.073700 | 0.081315 | 0.603657 | 0.842263 | 0.667390 | 0.562762 |
| 4 | 0.039800 | 0.083440 | 0.630992 | 0.858493 | 0.659339 | 0.608502 |
| 5 | 0.029000 | 0.089386 | 0.634223 | 0.859562 | 0.637766 | 0.634855 |
| 6 | 0.025700 | 0.092751 | 0.645494 | 0.860553 | 0.637458 | 0.661067 |
| 7 | 0.018100 | 0.096551 | 0.667813 | 0.857308 | 0.705766 | 0.680593 |
| 8 | 0.014400 | 0.097034 | 0.671131 | 0.858389 | 0.726101 | 0.663730 |

Figura 5.4.1.1.4 Resultados RoBERTa -base EP

En la Figura 5.4.1.1.4 se muestra la evolución de las métricas del modelo roberta-base a lo largo de las distintas épocas de entrenamiento. Se observa una disminución progresiva tanto de la training loss como de la validation loss, lo que indica un proceso de entrenamiento estable y una adecuada capacidad de generalización.

El análisis del F1 Macro muestra que el mejor rendimiento se alcanza en la época 8, con un valor de 0.671131.

| Modelo | Mejor época | F1 Macro |
|----------------------------|-------------|----------|
| BERT-base-uncased | 6 | 0.6278 |
| DeBERTa-v3-base | 8 | 0.5974 |
| ELECTRA-base-discriminator | 8 | 0.5892 |
| RoBERTa-base | 8 | 0.6711 |

Figura 5.4.1.1.5 Comparación de modelos encoder-only en la tarea EP

Como se puede observar en la Figura 5.4.1.1.5, el modelo que ha alcanzado el mejor rendimiento en términos de F1 Macro ha sido RoBERTa, por lo que se ha seleccionado como modelo base para los entrenamientos posteriores.

Es importante destacar que este primer entrenamiento se ha realizado utilizando la versión RoBERTa-base. Algunos modelos, como RoBERTa, disponen de variantes de mayor tamaño y capacidad; en este caso, RoBERTa-large, que cuenta con un número significativamente superior de parámetros y, por tanto, tiene el potencial de ofrecer un mejor rendimiento. Sin embargo, esta mejora conlleva un incremento notable en el consumo de memoria y en el tiempo de entrenamiento.

Para ejecutar el modelo RoBERTa-large fue necesario cambiar a una GPU más potente proporcionada por Google Colab. Aun así, el coste computacional resultó considerablemente mayor: mientras que los entrenamientos con RoBERTa-base se completaban en aproximadamente 20 minutos, la misma ejecución con RoBERTa-large requirió alrededor de dos horas, manteniendo constantes el resto de los parámetros.

Aunque RoBERTa-large mostró una ligera mejora en la métrica F1 Macro, esta no fue lo suficientemente significativa como para justificar el elevado incremento en recursos computacionales y tiempo de ejecución en el contexto de este trabajo. Por este motivo, se optó por continuar utilizando RoBERTa-base como modelo principal.

5.4.1.2 Comparación de entrenamientos Encoder-only Transformer (EF)

En esta sección se presentan los resultados obtenidos a partir de los entrenamientos orientados a la tarea de etiquetación efectiva (EF). Para ello, se han evaluado cuatro modelos basados en arquitecturas encoder-only Transformer en sus versiones base, con el objetivo de identificar el modelo que ofrece el mejor rendimiento.

Al igual que en el apartado de anotación epistemic, se ha priorizado el uso del F1 Macro como métrica principal de evaluación, debido al posible desbalanceo entre clases.

A continuación, se muestran los resultados obtenidos durante el entrenamiento del modelo **bert-base-uncased** para la tarea de etiquetación efectiva.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.131900 | 0.124284 | 0.188442 | 0.741327 | 0.188044 | 0.189592 |
| 2 | 0.094800 | 0.095990 | 0.421162 | 0.805795 | 0.556870 | 0.376319 |
| 3 | 0.064800 | 0.088081 | 0.516142 | 0.819164 | 0.617862 | 0.462427 |
| 4 | 0.042100 | 0.088829 | 0.556467 | 0.831302 | 0.598725 | 0.529053 |
| 5 | 0.035700 | 0.096864 | 0.575309 | 0.818868 | 0.571893 | 0.582209 |
| 6 | 0.016000 | 0.096982 | 0.583290 | 0.832843 | 0.582752 | 0.587332 |
| 7 | 0.015900 | 0.098141 | 0.578953 | 0.832975 | 0.583317 | 0.576322 |
| 8 | 0.010200 | 0.098732 | 0.584737 | 0.836860 | 0.592300 | 0.580601 |

Figura 5.4.1.2.1 Resultados bert-base-uncased EF

En la Figura 5.4.1.2.1 se muestra la evolución de las métricas del modelo bert-base-uncased a lo largo de las distintas épocas de entrenamiento. Se observa una disminución progresiva del training loss, mientras que la validation loss se mantiene relativamente estable.

El análisis del F1 Macro indica que el mejor rendimiento se alcanza en la época 8, con un valor de 0.584737. Dado que no se observan mejoras significativas en épocas posteriores, se selecciona dicha época como la mejor configuración del modelo para esta tarea.

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo **microsoft/deberta-v3-base** para la tarea de etiquetación efectiva.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.136800 | 0.126792 | 0.118753 | 0.703704 | 0.107430 | 0.133943 |
| 2 | 0.102100 | 0.092378 | 0.349553 | 0.801825 | 0.405074 | 0.321048 |
| 3 | 0.077400 | 0.079759 | 0.435403 | 0.824087 | 0.487442 | 0.404187 |
| 4 | 0.058100 | 0.081193 | 0.514100 | 0.844541 | 0.618780 | 0.482678 |
| 5 | 0.047600 | 0.084181 | 0.597257 | 0.849906 | 0.589181 | 0.607840 |
| 6 | 0.024700 | 0.088724 | 0.611950 | 0.861333 | 0.608231 | 0.620562 |
| 7 | 0.027300 | 0.088291 | 0.603496 | 0.861208 | 0.611235 | 0.602047 |
| 8 | 0.015200 | 0.089716 | 0.604867 | 0.863248 | 0.607658 | 0.605204 |

Figura 5.4.1.2.2 Resultados microsoft/deberta-v3-base EF

La Figura 5.4.1.2.2 recoge la evolución de las métricas del modelo microsoft/deberta-v3-base a lo largo del entrenamiento. El comportamiento

observado refleja una reducción progresiva de la training loss, mientras que la validation loss presenta ligeras fluctuaciones sin evidencias claras de sobreajuste.

En términos de rendimiento, el valor máximo de F1 Macro se alcanza en la época 6 (0.611950), sin observarse mejoras relevantes en iteraciones posteriores, por lo que esta configuración se considera la más adecuada para este modelo.

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo **electra-base-discriminator** para la tarea de etiquetación efectiva.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision | Macro Recall | Macro |
|-------|---------------|-----------------|----------|----------|-----------|--------------|-------|
| 1 | 0.143800 | 0.137737 | 0.089517 | 0.612095 | 0.099190 | 0.081714 | |
| 2 | 0.110200 | 0.099263 | 0.297736 | 0.793431 | 0.363938 | 0.269949 | |
| 3 | 0.081800 | 0.091526 | 0.354745 | 0.804488 | 0.417549 | 0.325959 | |
| 4 | 0.062600 | 0.096592 | 0.461664 | 0.818007 | 0.549632 | 0.417205 | |
| 5 | 0.058000 | 0.095924 | 0.512296 | 0.818356 | 0.533649 | 0.498735 | |
| 6 | 0.031100 | 0.101451 | 0.546062 | 0.826099 | 0.585890 | 0.533479 | |
| 7 | 0.032600 | 0.097172 | 0.571316 | 0.834688 | 0.599095 | 0.550473 | |
| 8 | 0.021800 | 0.101191 | 0.568126 | 0.832569 | 0.586601 | 0.556754 | |

Figura 5.4.1.2.3 Resultados electra-base-discriminator EF

En la Figura 5.4.1.2.3 se muestra la evolución de las métricas del modelo electra-base-discriminator a lo largo de las distintas épocas de entrenamiento. La training loss disminuye progresivamente, mientras que la validation loss se mantiene estable en las últimas épocas.

El análisis del F1 Macro indica que el mejor rendimiento se alcanza en la época 7, con un valor de 0.571316.

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo **roberta-base** para la tarea de etiquetación efectiva.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision | Macro Recall | Macro |
|-------|---------------|-----------------|----------|----------|-----------|--------------|-------|
| 1 | 0.117000 | 0.117155 | 0.205169 | 0.730040 | 0.236646 | 0.192406 | |
| 2 | 0.093600 | 0.089159 | 0.552474 | 0.830928 | 0.597591 | 0.521486 | |
| 3 | 0.065900 | 0.081342 | 0.550946 | 0.838409 | 0.609620 | 0.514668 | |
| 4 | 0.046700 | 0.081056 | 0.577157 | 0.848043 | 0.621026 | 0.546973 | |
| 5 | 0.040500 | 0.085690 | 0.589125 | 0.843425 | 0.570535 | 0.612660 | |
| 6 | 0.021200 | 0.087574 | 0.600105 | 0.855692 | 0.612364 | 0.592436 | |
| 7 | 0.020300 | 0.089362 | 0.598040 | 0.853691 | 0.609392 | 0.590174 | |
| 8 | 0.013400 | 0.090072 | 0.601936 | 0.854933 | 0.605007 | 0.601891 | |

Figura 5.4.1.2.4 Resultados Roberta-base EF

En la Figura 5.4.1.2.4 se muestra la evolución de las métricas del modelo roberta-base a lo largo de las distintas épocas de entrenamiento. Se observa una disminución progresiva del training loss, mientras que la validation loss se mantiene estable, lo que indica un proceso de entrenamiento consistente.

El análisis del F1 Macro muestra que el mejor rendimiento se alcanza en la época 8, con un valor de 0.601936.

| Modelo | Mejor época | F1 Macro |
|----------------------------|-------------|----------|
| BERT-base-uncased | 8 | 0.5847 |
| DeBERTa-v3-base | 6 | 0.6120 |
| ELECTRA-base-discriminator | 7 | 0.5713 |
| RoBERTa-base | 8 | 0.6019 |

Figura 5.4.1.2.5 Comparación de modelos encoder-only en la tarea EF

Como se observa en la Figura 5.4.1.2.5, el modelo microsoft/deberta-v3-base alcanza el mejor rendimiento global en términos de F1 Macro para la tarea EF. Aunque la diferencia con respecto a otros modelos no es especialmente pronunciada, este resultado justifica su selección como modelo base para los experimentos posteriores.

Al igual que ocurre en el caso de RoBERTa para la tarea EP, DeBERTa dispone de una variante de mayor tamaño (DeBERTa-large) que podría ofrecer mejoras adicionales en el rendimiento. No obstante, el incremento en el coste computacional y en los requisitos de memoria hace que su utilización se reserve para trabajos futuros o entornos con mayores recursos, como GPUs o TPUs de mayor capacidad.

5.4.1.3 Comparación de entrenamientos Encoder-only Transformer (EMO)

En esta sección se presentan los resultados obtenidos a partir de los entrenamientos orientados a la tarea de etiquetación emocional (EMO). A diferencia de las tareas EP y EF, esta tarea se encuentra todavía en una fase más experimental, ya que se dispone de una menor cantidad de datos anotados, lo que puede afectar a la estabilidad y fiabilidad de los resultados obtenidos.

Al igual que en los apartados anteriores, se han evaluado cuatro modelos basados en arquitecturas encoder-only Transformer en sus versiones base, utilizando el F1 Macro como métrica principal de evaluación.

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo **bert-base-uncased** para la tarea de etiquetación emocional.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision | Macro Recall | Macro |
|-------|---------------|-----------------|----------|----------|-----------|--------------|----------|
| 1 | 0.139000 | 0.108481 | 0.044380 | 0.740807 | 0.044380 | 0.044380 | 0.044380 |
| 2 | 0.088200 | 0.081868 | 0.044708 | 0.747547 | 0.044395 | 0.045025 | |
| 3 | 0.071700 | 0.070087 | 0.044813 | 0.748883 | 0.044603 | 0.045025 | |
| 4 | 0.059700 | 0.061511 | 0.085970 | 0.761062 | 0.087314 | 0.084702 | |
| 5 | 0.056100 | 0.053997 | 0.114191 | 0.772329 | 0.178293 | 0.105417 | |
| 6 | 0.042400 | 0.048697 | 0.312557 | 0.820084 | 0.510339 | 0.257503 | |
| 7 | 0.037500 | 0.046197 | 0.349375 | 0.832093 | 0.517830 | 0.300258 | |
| 8 | 0.037000 | 0.045294 | 0.419823 | 0.854604 | 0.604135 | 0.357346 | |

Figura 5.4.1.3.1 Resultados bert-base-uncased EMO

En la Figura 5.4.1.3.1 se muestra la evolución de las métricas del modelo a lo largo de las distintas épocas de entrenamiento. Se observa una disminución progresiva de la *training loss*, mientras que la *validation loss* se mantiene relativamente estable, lo que indica un proceso de entrenamiento consistente.

El análisis del F1 Macro indica que el mejor rendimiento se alcanza en la época 8, con un valor de 0.419823

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo **microsoft/deberta-v3-base** para la tarea de etiquetación emocional.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.103900 | 0.082929 | 0.044432 | 0.741472 | 0.044485 | 0.044380 |
| 2 | 0.081700 | 0.078012 | 0.044760 | 0.748214 | 0.044499 | 0.045025 |
| 3 | 0.075600 | 0.073389 | 0.044816 | 0.749777 | 0.044400 | 0.045240 |
| 4 | 0.067300 | 0.066055 | 0.044650 | 0.745520 | 0.044597 | 0.044702 |
| 5 | 0.070100 | 0.059699 | 0.073522 | 0.755319 | 0.083462 | 0.067859 |
| 6 | 0.056900 | 0.055676 | 0.161677 | 0.784179 | 0.205428 | 0.155554 |
| 7 | 0.049500 | 0.054098 | 0.185871 | 0.789831 | 0.184915 | 0.187232 |
| 8 | 0.051000 | 0.052720 | 0.192932 | 0.794567 | 0.193131 | 0.193725 |

Figura 5.4.1.3.2 Resultados microsoft/deberta-v3-base EMO

La evolución de las métricas del modelo microsoft/deberta-v3-base, mostrada en la Figura 5.4.1.3.2, indica que, aunque tanto la training loss como la validation loss disminuyen progresivamente a lo largo de las épocas, el valor de F1 Macro permanece bajo durante la mayor parte del entrenamiento, alcanzando su máximo en la época 8, con un valor de 0.1929.

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo **electra-base-discriminator** para la tarea de etiquetación emocional.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.148800 | 0.105094 | 0.044601 | 0.745763 | 0.044289 | 0.044917 |
| 2 | 0.083700 | 0.080587 | 0.044377 | 0.740341 | 0.044483 | 0.044273 |
| 3 | 0.075100 | 0.073749 | 0.044708 | 0.747547 | 0.044395 | 0.045025 |
| 4 | 0.064000 | 0.064095 | 0.044814 | 0.749331 | 0.044501 | 0.045132 |
| 5 | 0.062700 | 0.056479 | 0.044760 | 0.748214 | 0.044499 | 0.045025 |
| 6 | 0.049500 | 0.052007 | 0.175392 | 0.779690 | 0.196913 | 0.165163 |
| 7 | 0.043800 | 0.048300 | 0.254656 | 0.805369 | 0.282154 | 0.236802 |
| 8 | 0.042400 | 0.047400 | 0.310113 | 0.818859 | 0.369891 | 0.294133 |

Figura 5.4.1.3.3 Resultados electra-base-discriminator EMO

En la Figura 5.4.1.3.3 se muestra la evolución de las métricas del modelo **electra-base-discriminator** a lo largo de las distintas épocas de entrenamiento. Se observa una disminución progresiva tanto de la training loss como de la validation loss, lo que indica un proceso de entrenamiento estable y una correcta convergencia del modelo.

El análisis del F1 Macro indica que el mejor rendimiento se alcanza en la época 8, con un valor de 0.3101.

A continuación, se presentan los resultados obtenidos durante el entrenamiento del modelo Roberta-base para la tarea de etiquetación emocional.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.118500 | 0.095447 | 0.044711 | 0.748444 | 0.044195 | 0.045240 |
| 2 | 0.079000 | 0.075943 | 0.044156 | 0.735798 | 0.044474 | 0.043843 |
| 3 | 0.063300 | 0.060739 | 0.118101 | 0.773356 | 0.112446 | 0.124785 |
| 4 | 0.048400 | 0.049499 | 0.279735 | 0.816667 | 0.426826 | 0.259118 |
| 5 | 0.043500 | 0.044339 | 0.440315 | 0.859608 | 0.511152 | 0.416677 |
| 6 | 0.035100 | 0.040207 | 0.511260 | 0.884969 | 0.556148 | 0.487001 |
| 7 | 0.028700 | 0.037550 | 0.563226 | 0.895659 | 0.612298 | 0.531138 |
| 8 | 0.026600 | 0.037377 | 0.592901 | 0.902736 | 0.663985 | 0.551476 |

Figura 5.4.1.3.4 Resultados Roberta-base EMO para comparar

En la Figura 5.4.1.3.4 se observa la evolución de las métricas del modelo *roberta-base* a lo largo de las distintas épocas de entrenamiento. Al igual que en los modelos anteriores, la *training loss* y la *validation loss* muestran una tendencia descendente, indicando un proceso de entrenamiento estable.

El análisis del F1 Macro indica que el mejor rendimiento se alcanza en la época 8, con un valor de 0.5929.

| Modelo | Mejor época | F1 Macro |
|----------------------------|-------------|----------|
| BERT-base-uncased | 8 | 0.4198 |
| DeBERTa-v3-base | 8 | 0.1929 |
| ELECTRA-base-discriminator | 8 | 0.3101 |
| RoBERTa-base | 8 | 0.5929 |

Figura 5.4.1.3.5 Comparación de modelos encoder-only en la tarea EMO

Como se observa en la Figura 5.4.1.3.5, el modelo *roberta-base* alcanza el mejor rendimiento para la tarea de etiquetación emocional entre las arquitecturas encoder-only evaluadas.

Al igual que en los apartados anteriores, existe una variante de mayor tamaño (*roberta-large*) que podría ofrecer mejoras adicionales en términos de rendimiento. No obstante, el incremento asociado en el coste computacional y en los requisitos de memoria hace que su utilización se reserve para trabajos futuros desarrollados en entornos con mayores recursos de cálculo.

Una vez ejecutados todos se muestra un resumen de los resultados:

| Modelo | EP (F1 Macro) | EF (F1 Macro) | EMO (F1 Macro) |
|----------------------------|---------------|---------------|----------------|
| BERT-base-uncased | 0.6278 | 0.5847 | 0.4198 |
| DeBERTa-v3-base | 0.5974 | 0.6120 | 0.1929 |
| ELECTRA-base-discriminator | 0.5892 | 0.5713 | 0.3101 |
| RoBERTa-base | 0.6711 | 0.6019 | 0.5929 |

Figura 5.4.1.3.6 Resultados

Como se observa en la Figura 5.4.1.3.6, el rendimiento de los modelos varía en función de la tarea considerada. RoBERTa-base obtiene los mejores resultados en las tareas EP y EMO, mientras que DeBERTa-v3-base alcanza el valor más alto en la tarea EF. Estos resultados ponen de manifiesto que no existe un único modelo óptimo para todas las tareas, aunque RoBERTa-base presenta el mejor rendimiento global, lo que justifica su selección como modelo de referencia para los experimentos posteriores.

Las diferencias observadas en el rendimiento de los modelos entre las tareas EP, EF y EMO pueden deberse, en gran medida, al tamaño y a la distribución de los conjuntos de datos empleados en cada caso. En particular, las tareas EP y EF cuentan con un mayor número de ejemplos anotados, lo que permite a los modelos aprender patrones más representativos y estables. Por el contrario, la

1

tarea EMO dispone de un conjunto de datos más reducido, lo que limita la capacidad de generalización de los modelos y se traduce en valores de F1 Macro inferiores.

5.4.1.4 Elección de los mejores hiperparámetros

Una vez identificado el modelo con mejor rendimiento para la tarea de etiquetación epistémica (EP), en este caso **RoBERTa-base**, el siguiente paso consistió en la optimización de sus hiperparámetros con el objetivo de maximizar el rendimiento del modelo. Los hiperparámetros controlan aspectos fundamentales del proceso de entrenamiento —como la tasa de aprendizaje o el número de épocas— y, a diferencia de los pesos del modelo, no se ajustan automáticamente durante el proceso de entrenamiento.

La optimización de hiperparámetros puede abordarse de forma manual, mediante la modificación iterativa de valores y la evaluación de los resultados obtenidos. No obstante, este enfoque resulta poco eficiente y difícilmente reproducible. Alternativamente, pueden emplearse métodos de búsqueda exhaustiva como **Grid Search**, que evalúan todas las combinaciones posibles de un conjunto predefinido de hiperparámetros, si bien su coste computacional es muy elevado, especialmente en modelos de gran tamaño como RoBERTa.

Por este motivo, en el presente trabajo se optó por el uso de **Optuna**, una librería en Python diseñada para la optimización automática de hiperparámetros en modelos de *machine learning* y *deep learning*. Optuna permite explorar el espacio de búsqueda de forma adaptativa, priorizando aquellas configuraciones que muestran un mejor rendimiento en función de una métrica objetivo, lo que reduce significativamente el tiempo de entrenamiento necesario y mejora la eficiencia del proceso de optimización.

La optimización de hiperparámetros se llevó a cabo mediante la ejecución de 20 trials, donde cada trial corresponde a un entrenamiento completo del modelo utilizando una combinación específica de hiperparámetros propuesta por Optuna. Tras cada trial, el modelo fue evaluado empleando la métrica F1 Macro, cuyo valor se utilizó como señal para guiar las exploraciones posteriores.

Aunque Optuna no transfiere el aprendizaje del modelo entre trials, sí ajusta progresivamente la exploración del espacio de hiperparámetros, priorizando aquellas regiones que han demostrado ofrecer mejores resultados. Además, la librería incorpora un mecanismo de *pruning*, que permite detener de forma anticipada aquellos entrenamientos cuyo rendimiento inicial es claramente inferior al de los mejores trials previos, reduciendo así el consumo de tiempo y recursos computacionales.

Optuna en RoBERTa-EP

A continuación, se muestra un extracto representativo de la salida de Optuna. En cada trial se prueba una configuración distinta, y se registra el valor final de la métrica objetivo F1 Macro. En algunos casos, Optuna detiene la ejecución anticipadamente mediante el mecanismo de pruning (por ejemplo, en los trials 6 y 19). Esto ocurre cuando, tras pocas épocas, el rendimiento es

3

3
3

56

13

10

claramente inferior al de los mejores trials anteriores y se estima que no será competitivo, por lo que se cancela el entrenamiento para ahorrar tiempo y recursos.

Etiquetas del modelo:

[1] ['EPAPH', 'EPAPH_P', 'EP_CGA', 'EP_CGA_P', 'EP_EM', 'EP_FV', 'EP_FV_P',
'EP_IIE', 'EP_IIE_P', 'EP_IIR', 'EP_MEP_P', 'NO_EP']

Map: 100%

8030/8030 [00:00<00:00, 23424.29 examples/s]

| [12848/12848 10:25, Epoch 16/16] | | | | | | |
|----------------------------------|---------------|-----------------|----------|----------|-----------------|--------------|
| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
| 1 | 0.705000 | 0.839676 | 0.317269 | 0.553137 | 0.230433 | 0.769589 |
| 2 | 0.766500 | 0.607144 | 0.511933 | 0.755633 | 0.417197 | 0.817349 |
| 3 | 0.533100 | 0.628023 | 0.693703 | 0.836433 | 0.624098 | 0.807175 |
| 4 | 0.314500 | 0.719304 | 0.663085 | 0.837503 | 0.582294 | 0.819655 |
| 5 | 0.248300 | 0.750831 | 0.700881 | 0.846261 | 0.619664 | 0.833153 |
| 6 | 0.184600 | 1.081570 | 0.738007 | 0.868876 | 0.716251 | 0.781579 |
| 7 | 0.092900 | 0.928752 | 0.725301 | 0.852404 | 0.660752 | 0.827340 |
| 8 | 0.079500 | 0.930727 | 0.738154 | 0.858032 | 0.683474 | 0.832631 |
| 9 | 0.045300 | 0.990834 | 0.752977 | 0.860386 | 0.700403 | 0.836089 |
| 10 | 0.026400 | 1.232408 | 0.750824 | 0.862364 | 0.724836 | 0.801788 |
| 11 | 0.025700 | 1.212561 | 0.735612 | 0.862265 | 0.705407 | 0.777983 |
| 12 | 0.021900 | 1.211783 | 0.762965 | 0.863025 | 0.739307 | 0.798776 |
| 13 | 0.013600 | 1.160532 | 0.792122 | 0.868519 | 0.777867 | 0.813242 |
| 14 | 0.046400 | 1.220665 | 0.778139 | 0.868670 | 0.765928 | 0.807333 |
| 15 | 0.021900 | 1.125011 | 0.786024 | 0.870986 | 0.769518 | 0.815755 |
| 16 | 0.007100 | 1.146237 | 0.777756 | 0.867849 | 0.763734 | 0.808838 |

[1 2025-12-31 12:17:50,813] Trial 0 finished with value: 0.7777556153425119 and parameters: {'learning_rate': 1.5101109926919294e-05, 'num_train_epochs': 16, 'per_device_train_batch_size': 8, 'weight_decay': 0.05378523743967753, 'warmup_ratio': 0.05718737793255753}.

| [24090/24090 18:49, Epoch 15/15] | | | | | | |
|----------------------------------|---------------|-----------------|----------|----------|-----------------|--------------|
| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
| 1 | 1.019300 | 1.178832 | 0.500422 | 0.759851 | 0.437876 | 0.657131 |
| 2 | 0.984900 | 1.303120 | 0.576932 | 0.813218 | 0.530558 | 0.644808 |
| 3 | 1.811900 | 1.150194 | 0.608948 | 0.815870 | 0.549982 | 0.708564 |
| 4 | 0.617500 | 1.333612 | 0.659995 | 0.832811 | 0.646362 | 0.695567 |
| 5 | 1.105000 | 1.419172 | 0.682105 | 0.843182 | 0.663965 | 0.727389 |
| 6 | 0.888200 | 1.464125 | 0.716060 | 0.853127 | 0.715358 | 0.750624 |
| 7 | 0.858300 | 1.511703 | 0.702280 | 0.848311 | 0.735307 | 0.749518 |
| 8 | 0.345400 | 1.216743 | 0.694649 | 0.844156 | 0.653920 | 0.779279 |
| 9 | 0.150800 | 1.479489 | 0.733691 | 0.849542 | 0.732997 | 0.794234 |
| 10 | 0.039900 | 2.167371 | 0.658571 | 0.852958 | 0.698569 | 0.669269 |
| 11 | 0.074900 | 1.296532 | 0.723576 | 0.850457 | 0.689979 | 0.800522 |
| 12 | 0.063800 | 1.541766 | 0.732415 | 0.851926 | 0.705197 | 0.791096 |
| 13 | 0.075700 | 1.740766 | 0.746471 | 0.853287 | 0.767158 | 0.763985 |
| 14 | 0.025200 | 1.564799 | 0.771448 | 0.857388 | 0.782015 | 0.802401 |
| 15 | 0.003800 | 1.548068 | 0.779746 | 0.854358 | 0.782019 | 0.804273 |

[1 2025-12-31 12:36:41,773] Trial 1 finished with value: 0.7797462437234873 and parameters: {'learning_rate': 5.271057027742492e-05, 'num_train_epochs': 15, 'per_device_train_batch_size': 4, 'weight_decay': 0.0794706140677424, 'warmup_ratio': 0.05552898075979318}.

Figura 5.4.1.4.1 Resultados ejecución optuna EP

[5226/5226 04:46, Epoch 13/13]

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.938300 | 0.916089 | 0.277523 | 0.485016 | 0.197020 | 0.779668 |
| 2 | 0.495700 | 0.489852 | 0.548565 | 0.758722 | 0.439976 | 0.884528 |
| 3 | 0.361100 | 0.462855 | 0.643569 | 0.784304 | 0.522047 | 0.892147 |
| 4 | 0.203900 | 0.620423 | 0.687715 | 0.838095 | 0.603052 | 0.845457 |
| 5 | 0.164400 | 0.616125 | 0.723376 | 0.849764 | 0.639794 | 0.863656 |
| 6 | 0.119300 | 0.794837 | 0.472806 | 0.864094 | 0.680873 | 0.846087 |
| 7 | 0.181700 | 0.756343 | 0.745108 | 0.841749 | 0.673321 | 0.858817 |
| 8 | 0.844600 | 0.886987 | 0.779949 | 0.864743 | 0.741413 | 0.826197 |
| 9 | 0.847300 | 0.897945 | 0.753684 | 0.871186 | 0.714476 | 0.826872 |
| 10 | 0.843800 | 1.188810 | 0.752661 | 0.865401 | 0.764818 | 0.787559 |
| 11 | 0.020000 | 0.944807 | 0.774336 | 0.867860 | 0.736542 | 0.828364 |
| 12 | 0.014600 | 0.994898 | 0.762967 | 0.867614 | 0.746390 | 0.816805 |
| 13 | 0.008400 | 1.014165 | 0.758691 | 0.866818 | 0.727539 | 0.814971 |

[I 2025-12-31 12:41:30,034] Trial 2 finished with value: 0.7586907453130881 and parameters: {'learning_rate': 3.5335592985849544e-05, 'num_train_epochs': 13, 'per_device_train_batch_size': 16, 'weight_decay': 0.008591413121306934, 'warmup_ratio': 0.12161053540564098}.

[1608/1608 01:24, Epoch 4/4]

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 0.676600 | 0.753257 | 0.383893 | 0.573443 | 0.285220 | 0.811659 |
| 2 | 0.411900 | 0.461436 | 0.579487 | 0.776399 | 0.462798 | 0.889012 |
| 3 | 0.295600 | 0.435533 | 0.651872 | 0.816948 | 0.540963 | 0.881433 |
| 4 | 0.156300 | 0.499738 | 0.709028 | 0.840738 | 0.612804 | 0.859835 |

[I 2025-12-31 13:09:26,464] Trial 5 finished with value: 0.7090203478678908 and parameters: {'learning_rate': 3.405312769146432e-05, 'num_train_epochs': 4, 'per_device_train_batch_size': 16, 'weight_decay': 0.05075824075266691, 'warmup_ratio': 0.05183660454563402}.

[402/6432 00:21 < 05:18, 18.96 it/s, Epoch 1/16]

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 1.241600 | 1.207690 | 0.211993 | 0.379956 | 0.167867 | 0.588612 |

[I 2025-12-31 13:09:48,416] Trial 6 pruned.

Some weights of RobertaForSequenceClassification were not initialized from the model checkpoint at roberta-base and are newly initialized: ['classifier.dense.bias', 'classifier.dense.weight', 'classifier.out_proj.bias', 'classifier.out_proj.weight']

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|
| 1 | 1.085200 | 0.998773 | 0.234606 | 0.410356 | 0.165145 | 0.782373 |

[I 2025-12-31 13:57:41,674] Trial 19 pruned.

Figura 5.4.1.4.2 Más resultados ejecución optuna EP

En las Figuras 5.4.1.4.1 y 5.4.1.4.2 se muestra un extracto representativo de la salida generada durante la ejecución de Optuna para la tarea EP, donde se observan distintos trials junto con sus configuraciones de hiperparámetros y los valores obtenidos de la métrica objetivo.

Una vez finalizado el proceso de optimización, Optuna identifica automáticamente el trial que ha alcanzado el mejor valor de F1 Macro, denominado *best run*. En este caso, el mejor resultado obtenido corresponde a un valor de F1 Macro = 0.7921, asociado a la configuración de hiperparámetros mostrada a continuación:

92 ==== BEST RUN ====

71 BestRun(run_id='0', objective= 0.792122, hyperparameters={'learning_rate':
48 1.5101109926919294e-05, 'num_train_epochs': 16,
 'per_device_train_batch_size': 8, 'weight_decay': 0.05378523743967753,
 'warmup_ratio': 0.05718737793255753}, run_summary=None)

Adicionalmente, al finalizar la ejecución, se genera automáticamente una carpeta denominada Best_hyperparameters, que contiene un archivo en formato JSON con la configuración óptima identificada, tal y como se muestra en la Figura 5.4.1.4.3.

```
{  
    "learning_rate": 1.5101109926919294e-05,  
    "num_train_epochs": 16,  
    "per_device_train_batch_size": 8,  
    "weight_decay": 0.05378523743967753,  
    "warmup_ratio": 0.05718737793255753  
}
```

Figura 5.4.1.4.3 Best_hyperparameters EP

Este archivo recoge los valores finales seleccionados para el entrenamiento del modelo, entre los que se incluyen los siguientes con una pequeña explicación [28]:

- **Learning rate:** tasa de aprendizaje, que determina la velocidad a la que el modelo actualiza sus pesos en cada paso del entrenamiento.
- **Num_train_epochs:** número total de épocas, es decir, el número de veces que el modelo recorre completamente el conjunto de datos de entrenamiento.
- **Per_device_train_batch_size:** número de ejemplos procesados simultáneamente en cada GPU antes de realizar la actualización de los gradientes.
- **Weight_decay:** término de regularización L2, utilizado para penalizar valores de pesos excesivamente grandes y reducir el riesgo de sobreajuste.
- **Warmup_ratio:** proporción inicial del entrenamiento durante la cual la tasa de aprendizaje aumenta de forma progresiva hasta alcanzar su valor máximo, lo que contribuye a una mayor estabilidad en las primeras fases del entrenamiento.

Una vez identificados los mejores hiperparámetros, estos se incorporan al script de entrenamiento definitivo, con el que se entrena el modelo final.

Optuna en RoBERTa-EF

Una vez descrito el proceso de optimización de hiperparámetros para la tarea epistémica (EP), se aplicó el mismo procedimiento a la tarea de etiquetación

84
1

efectiva (EF), utilizando igualmente el modelo DeBERTa-base como arquitectura de referencia. Dado que la metodología empleada es idéntica a la descrita en el apartado anterior, en esta sección se presentan únicamente los resultados correspondientes al mejor *trial* identificado por Optuna.

| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro | Macro |
|-------|---------------|-----------------|----------|----------|-----------------|--------------|-------|
| 1 | 0.793600 | 0.715073 | 0.346292 | 0.578545 | 0.256748 | 0.732474 | |
| 2 | 0.718500 | 0.619275 | 0.493581 | 0.729895 | 0.385512 | 0.776515 | |
| 3 | 0.547700 | 0.725973 | 0.579888 | 0.788991 | 0.487884 | 0.744793 | |
| 4 | 0.359800 | 0.932901 | 0.629608 | 0.828729 | 0.575735 | 0.711879 | |
| 5 | 0.303000 | 0.861143 | 0.610156 | 0.800491 | 0.538623 | 0.732042 | |
| 6 | 0.187900 | 0.972568 | 0.648347 | 0.835259 | 0.584915 | 0.737959 | |
| 7 | 0.088300 | 1.113456 | 0.662832 | 0.855317 | 0.621920 | 0.715509 | |
| 8 | 0.053900 | 1.263004 | 0.654414 | 0.856335 | 0.611729 | 0.708120 | |
| 9 | 0.041600 | 1.230669 | 0.681679 | 0.859794 | 0.656237 | 0.725531 | |
| 10 | 0.045600 | 1.149387 | 0.671244 | 0.850450 | 0.626559 | 0.734376 | |
| 11 | 0.023100 | 1.342168 | 0.684001 | 0.859371 | 0.652429 | 0.723729 | |
| 12 | 0.039200 | 1.306124 | 0.680835 | 0.861801 | 0.643116 | 0.727735 | |
| 13 | 0.049500 | 1.433073 | 0.670868 | 0.860098 | 0.634326 | 0.714740 | |
| 14 | 0.013300 | 1.375425 | 0.669605 | 0.854630 | 0.629993 | 0.719023 | |
| 15 | 0.008100 | 1.407352 | 0.671656 | 0.858994 | 0.635888 | 0.715349 | |

[I 2025-12-31 15:56:45,760] Trial 9 finished with value: 0.6716556850602888 and parameters: {'learning_rate': 3.0358233863984693e-05, 'num_train_epochs': 15, 'per_device_train_batch_size': 16, 'weight_decay': 0.0028376159480514975, 'warmup_ratio': 0.007962962268492847}.

Figura 5.4.1.4.4 Resultados Mejor Trial EF

La Figura 5.4.1.4.4 muestra el resultado del mejor *trial* obtenido durante la optimización para la tarea EF. En este caso, el *trial* número 9 alcanzó el mejor rendimiento, con un valor de F1 Macro = 0.6840, correspondiente a la época 11 de entrenamiento.

Y el json que nos otorga como Best_hyperparameters es:

```
{  
    "learning_rate": 3.0358233863984693e-05,  
    "num_train_epochs": 15,  
    "per_device_train_batch_size": 16,  
    "weight_decay": 0.0028376159480514975,  
    "warmup_ratio": 0.007962962268492847  
}
```

Figura 5.4.1.4.5 Best_hyperparameters EF

En la Figura 5.4.1.4.5 se presenta el archivo Best_hyperparameters generado automáticamente por Optuna, en el que se recogen los valores óptimos de los hiperparámetros asociados al mejor *run* identificado para esta tarea. Estos hiperparámetros se utilizaron posteriormente en el entrenamiento definitivo del modelo DeBERTa -EF.

Optuna en RoBERTa-EMO

De forma análoga a los casos anteriores, se llevó a cabo la optimización de hiperparámetros para la tarea de etiquetación emocional (EMO), nuevamente empleando el modelo RoBERTa-base y la métrica F1 Macro como función objetivo.

| [5472/5472 04:51, Epoch 16/16] | | | | | | |
|--------------------------------|---------------|-----------------|----------|----------|-----------------|--------------|
| Epoch | Training Loss | Validation Loss | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
| 1 | 0.866100 | 0.623336 | 0.231753 | 0.404249 | 0.149145 | 0.924128 |
| 2 | 0.434400 | 0.393992 | 0.413212 | 0.621495 | 0.288742 | 0.926695 |
| 3 | 0.243800 | 0.405221 | 0.610046 | 0.804726 | 0.509366 | 0.865668 |
| 4 | 0.155900 | 0.373124 | 0.646683 | 0.823752 | 0.564882 | 0.891136 |
| 5 | 0.154400 | 0.385801 | 0.739753 | 0.887825 | 0.672223 | 0.878004 |
| 6 | 0.090000 | 0.486826 | 0.778737 | 0.900789 | 0.758016 | 0.838284 |
| 7 | 0.073600 | 0.453772 | 0.792553 | 0.909737 | 0.752624 | 0.869104 |
| 8 | 0.043900 | 0.603241 | 0.791385 | 0.916129 | 0.787351 | 0.831348 |
| 9 | 0.037700 | 0.591575 | 0.802734 | 0.919424 | 0.798958 | 0.841840 |
| 10 | 0.029600 | 0.551361 | 0.818171 | 0.921752 | 0.794581 | 0.872150 |
| 11 | 0.009000 | 0.576194 | 0.829783 | 0.929964 | 0.828746 | 0.854931 |
| 12 | 0.009000 | 0.562567 | 0.841793 | 0.932471 | 0.830978 | 0.866272 |
| 13 | 0.008400 | 0.604216 | 0.823255 | 0.921526 | 0.812074 | 0.855433 |
| 14 | 0.005900 | 0.607042 | 0.830281 | 0.931685 | 0.819033 | 0.862966 |
| 15 | 0.006600 | 0.642667 | 0.829421 | 0.929191 | 0.824365 | 0.858848 |
| 16 | 0.005800 | 0.640136 | 0.832168 | 0.929964 | 0.825904 | 0.861374 |

[I 2026-01-01 17:56:13,368] Trial 4 finished with value: 0.8321679296821937 and parameters: {'learning_rate': 3.446319711001856e-05, 'num_train_epochs': 16, 'per_device_train_batch_size': 8, 'weight_decay': 0.04140322769361629, 'warmup_ratio': 0.06296962684244688}.

Figura 5.4.1.4.6 Resultados Best run EMO

La Figura 5.4.1.4.6 muestra el resultado correspondiente al mejor *trial* identificado por Optuna para esta tarea. En este caso, el *trial* número 4 obtuvo el mejor rendimiento, alcanzando un valor de F1 Macro = 0.8417 en la época 12.

Y el Json que nos otorga como Best_hyperparameters es:

```
{  
    "learning_rate": 3.446319711001856e-05,  
    "num_train_epochs": 16,  
    "per_device_train_batch_size": 8,  
    "weight_decay": 0.04140322769361629,  
    "warmup_ratio": 0.06296962684244688  
}
```

Figura 5.4.1.4.7 Best_hyperparameters EMO

El archivo Best_hyperparameters asociado a este *trial* se presenta en la Figura 5.4.1.4.7, donde se detallan los valores óptimos de los hiperparámetros seleccionados automáticamente por Optuna para el entrenamiento final del modelo RoBERTa-EMO.

| Tarea / Modelo | Mejor Trial | F1 Macro | Epoch óptima |
|----------------|-------------|----------|--------------|
| RoBERTa-EP | Trial 0 | 0.7921 | 16 |
| DeBERTa-v3-EF | Trial 9 | 0.6840 | 11 |
| RoBERTa-EMO | Trial 4 | 0.8410 | 12 |

Figura 5.4.1.4.8 Resumen optuna mejores trials

La Figura 5.4.1.4.8 presenta un resumen global de los mejores *trials* obtenidos mediante Optuna para las tres tareas consideradas: EP, EF y EMO. Este resumen permite observar de forma conjunta los valores máximos alcanzados de la métrica F1 Macro y las configuraciones de hiperparámetros seleccionadas en cada caso.

Una vez completado el proceso de optimización y obtenidos los hiperparámetros óptimos para cada tarea, el siguiente paso consistió en realizar los entrenamientos definitivos de los modelos correspondientes, utilizando dichas configuraciones como base para su integración en el sistema final.

5.4.1.5 Resultados inicio/final

Una vez identificados los modelos con mejor rendimiento para cada una de las tareas de clasificación y seleccionadas las configuraciones óptimas de hiperparámetros, se procedió a la ejecución de los entrenamientos finales. El objetivo de esta fase fue analizar el impacto del tamaño del conjunto de entrenamiento y del uso de un conjunto de test explícito en el rendimiento de los modelos.

En particular, se compararon dos configuraciones de entrenamiento:

1. una partición entrenamiento–validación–test (80–10–10)
2. una partición entrenamiento–validación (80–20) sin conjunto de test explícito.

En todos los casos se emplearon como modelos de referencia aquellos que habían mostrado el mejor rendimiento en las comparaciones previas.

En primer lugar, se analizan los resultados obtenidos para la tarea de clasificación epistémica (EP), empleando el modelo RoBERTa-base como

referencia.

| <u>Epoch</u> | <u>Training Loss</u> | <u>Validation Loss</u> | F1 Macro | F1 Micro | Precision | Macro | Recall | Macro |
|--------------|----------------------|------------------------|----------|----------|-----------|----------|--------|-------|
| 1 | 0.138700 | 0.126624 | 0.216464 | 0.767518 | 0.301074 | 0.200127 | | |
| 2 | 0.070500 | 0.089388 | 0.414211 | 0.839853 | 0.533657 | 0.374086 | | |
| 3 | 0.072900 | 0.076592 | 0.566672 | 0.854080 | 0.619890 | 0.529227 | | |
| 4 | 0.044900 | 0.077373 | 0.601996 | 0.862884 | 0.625540 | 0.582239 | | |
| 5 | 0.043300 | 0.076729 | 0.636138 | 0.870588 | 0.711152 | 0.615485 | | |
| 6 | 0.030000 | 0.080695 | 0.653765 | 0.871252 | 0.693349 | 0.650294 | | |
| 7 | 0.018700 | 0.090366 | 0.683548 | 0.865116 | 0.730113 | 0.674545 | | |
| 8 | 0.010500 | 0.095904 | 0.726270 | 0.871137 | 0.882511 | 0.680495 | | |
| 9 | 0.010800 | 0.097369 | 0.734055 | 0.867946 | 0.865229 | 0.695307 | | |
| 10 | 0.005900 | 0.096911 | 0.682410 | 0.866511 | 0.782174 | 0.657292 | | |
| 11 | 0.008500 | 0.096179 | 0.718409 | 0.876457 | 0.823726 | 0.713868 | | |
| 12 | 0.006500 | 0.098716 | 0.706747 | 0.873025 | 0.817574 | 0.693768 | | |
| 13 | 0.003100 | 0.099654 | 0.727680 | 0.874272 | 0.806492 | 0.725783 | | |
| 14 | 0.004400 | 0.104583 | 0.699089 | 0.869616 | 0.805795 | 0.688622 | | |
| 15 | 0.002800 | 0.105669 | 0.715815 | 0.877480 | 0.823051 | 0.705166 | | |
| 16 | 0.003200 | 0.105861 | 0.707820 | 0.875437 | 0.812563 | 0.700073 | | |

Figura 5.4.1.5.1 Resultados Definitivos EP con test

La Figura 5.4.1.5.1 muestra los resultados obtenidos al entrenar el modelo con una partición 80–10–10. En esta configuración, el modelo alcanza un valor final de F1 Macro del 74 % sobre el conjunto de test lo que indica una adecuada capacidad de generalización sobre datos no vistos durante el entrenamiento.

| <u>Epoch</u> | <u>Training Loss</u> | <u>Validation Loss</u> | F1 Macro | F1 Micro | Precision | Macro | Recall | Macro |
|--------------|----------------------|------------------------|----------|----------|-----------|----------|--------|-------|
| 1 | 0.630400 | 0.839676 | 0.317269 | 0.553137 | 0.230433 | 0.769589 | | |
| 2 | 0.560800 | 0.607548 | 0.512322 | 0.756569 | 0.417374 | 0.817184 | | |
| 3 | 0.775200 | 0.633835 | 0.700430 | 0.836942 | 0.638738 | 0.804376 | | |
| 4 | 0.204000 | 0.692602 | 0.648342 | 0.832604 | 0.566192 | 0.818581 | | |
| 5 | 0.217500 | 0.683932 | 0.720952 | 0.843767 | 0.637275 | 0.856291 | | |
| 6 | 0.295400 | 1.043960 | 0.749319 | 0.868138 | 0.729856 | 0.791120 | | |
| 7 | 0.136700 | 1.004378 | 0.747933 | 0.854396 | 0.702770 | 0.819750 | | |
| 8 | 0.060800 | 1.029118 | 0.741788 | 0.860867 | 0.701368 | 0.802938 | | |
| 9 | 0.066400 | 1.002651 | 0.737758 | 0.856738 | 0.677749 | 0.836862 | | |
| 10 | 0.023600 | 1.263793 | 0.754467 | 0.867449 | 0.748797 | 0.774293 | | |
| 11 | 0.030100 | 1.145426 | 0.743252 | 0.863572 | 0.701812 | 0.798150 | | |
| 12 | 0.009300 | 1.170412 | 0.766729 | 0.869292 | 0.748840 | 0.800779 | | |
| 13 | 0.007400 | 1.172029 | 0.782420 | 0.867601 | 0.751912 | 0.823177 | | |
| 14 | 0.019200 | 1.168676 | 0.752397 | 0.863649 | 0.729478 | 0.792708 | | |
| 15 | 0.016500 | 1.129208 | 0.778033 | 0.869119 | 0.745900 | 0.823104 | | |
| 16 | 0.005500 | 1.125582 | 0.777304 | 0.867065 | 0.744750 | 0.822332 | | |

Figura 5.4.1.5.2 Resultados Definitivos EP sin test

Por su parte, la Figura 5.4.1.5.2 presenta los resultados del entrenamiento sin conjunto de test explícito, empleando una partición 80–20. En este caso, se observa una mejora del rendimiento, con un incremento del F1 Macro desde 0.718 hasta 0.782. Este aumento puede atribuirse al uso de un mayor volumen de datos durante el entrenamiento, lo que permite al modelo aprender representaciones más robustas

Adicionalmente, la optimización de hiperparámetros mediante Optuna contribuyó de forma significativa a la mejora del rendimiento. Concretamente, el valor de F1 Macro aumentó desde 0.671 hasta 0.782, lo que supone una mejora superior al 10 %. Este resultado pone de manifiesto el impacto positivo de la optimización de hiperparámetros en modelos basados en arquitecturas Transformer. En segundo lugar, se analizan los resultados correspondientes a la tarea de clasificación efectiva (EF), utilizando el modelo microsoft/deberta-v3-base como referencia.

| <u>Epoch</u> | <u>Training Loss</u> | <u>Validation Loss</u> | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|--------------|----------------------|------------------------|----------|----------|-----------------|--------------|
| 1 | 0.104900 | 0.122756 | 0.184026 | 0.724537 | 0.173310 | 0.200485 |
| 2 | 0.080700 | 0.094051 | 0.323693 | 0.805094 | 0.366394 | 0.305885 |
| 3 | 0.064400 | 0.084912 | 0.523777 | 0.826964 | 0.561721 | 0.507087 |
| 4 | 0.043600 | 0.084093 | 0.617911 | 0.857293 | 0.647651 | 0.601246 |
| 5 | 0.045100 | 0.102804 | 0.600870 | 0.831519 | 0.616443 | 0.601618 |
| 6 | 0.027900 | 0.102334 | 0.619274 | 0.848837 | 0.637341 | 0.609119 |
| 7 | 0.019900 | 0.105992 | 0.609036 | 0.855050 | 0.627487 | 0.598787 |
| 8 | 0.017900 | 0.105173 | 0.618512 | 0.851599 | 0.598469 | 0.632238 |
| 9 | 0.015100 | 0.118726 | 0.593966 | 0.851042 | 0.587263 | 0.613055 |
| 10 | 0.017200 | 0.109911 | 0.622373 | 0.863136 | 0.646247 | 0.605275 |
| 11 | 0.011400 | 0.117467 | 0.605707 | 0.855503 | 0.601236 | 0.616116 |
| 12 | 0.010800 | 0.116977 | 0.613719 | 0.859833 | 0.610817 | 0.621253 |
| 13 | 0.006800 | 0.122924 | 0.615296 | 0.858633 | 0.610252 | 0.625155 |
| 14 | 0.002900 | 0.124848 | 0.615370 | 0.859969 | 0.609400 | 0.625088 |
| 15 | 0.002400 | 0.125454 | 0.618485 | 0.861890 | 0.614230 | 0.627774 |

Figura 5.4.1.5.3 Resultados Definitivos EF con test

La Figura 5.4.1.5.3 muestra los resultados obtenidos utilizando una partición 80–10–10. En este escenario, el modelo alcanza un valor de F1 Macro del 63 % sobre el conjunto de test, lo que evidencia un comportamiento consistente al evaluarse sobre datos no vistos.

| <u>Epoch</u> | <u>Training Loss</u> | <u>Validation Loss</u> | F1 Macro | F1 Micro | Precision Macro | Recall Macro |
|--------------|----------------------|------------------------|----------|----------|-----------------|--------------|
| 1 | 0.812300 | 0.857573 | 0.455262 | 0.686167 | 0.362936 | 0.666024 |
| 2 | 1.237800 | 0.846086 | 0.527635 | 0.775712 | 0.445660 | 0.691074 |
| 3 | 1.334700 | 1.010468 | 0.616523 | 0.791636 | 0.561567 | 0.719905 |
| 4 | 0.434800 | 1.138950 | 0.603137 | 0.814631 | 0.539624 | 0.698327 |
| 5 | 0.711500 | 1.159549 | 0.659402 | 0.849567 | 0.602292 | 0.738329 |
| 6 | 0.142500 | 1.265800 | 0.643344 | 0.837363 | 0.593608 | 0.709194 |
| 7 | 0.303400 | 1.598554 | 0.638792 | 0.853659 | 0.615724 | 0.675267 |
| 8 | 0.047400 | 1.446024 | 0.640711 | 0.847362 | 0.590404 | 0.710050 |
| 9 | 0.028700 | 1.601596 | 0.644224 | 0.850431 | 0.611570 | 0.684410 |
| 10 | 0.052900 | 1.695252 | 0.661410 | 0.852083 | 0.631431 | 0.700611 |
| 11 | 0.032400 | 1.781785 | 0.646944 | 0.853361 | 0.610487 | 0.690810 |
| 12 | 0.009000 | 1.774895 | 0.663008 | 0.854688 | 0.630560 | 0.702659 |
| 13 | 0.003900 | 2.009175 | 0.647309 | 0.849649 | 0.616524 | 0.689418 |
| 14 | 0.002100 | 2.037506 | 0.657475 | 0.862745 | 0.634477 | 0.685415 |
| 15 | 0.002800 | 2.021182 | 0.675385 | 0.860951 | 0.656944 | 0.700612 |
| 16 | 0.003600 | 1.989283 | 0.668555 | 0.858262 | 0.641703 | 0.701531 |

Figura 5.4.1.5.4 Resultados Definitivos EF sin test

En la Figura 5.4.1.5.4 se presentan los resultados del entrenamiento sin conjunto de test explícito, con una partición 80–20. En este caso, el F1 Macro aumenta de 0.612 a 0.675, reflejando nuevamente el efecto positivo del incremento del conjunto de entrenamiento.

Tras la optimización de hiperparámetros mediante Optuna, el modelo mejora su rendimiento desde 0.612 hasta 0.675. Aunque esta mejora es menos pronunciada que en la tarea EP, resulta coherente con el tamaño y la distribución del conjunto de datos disponible para la tarea EF.

Finalmente, se analizan los resultados correspondientes a la tarea de clasificación emocional (EMO), utilizando RoBERTa-base como modelo de referencia.

| <u>Epoch</u> | <u>Training Loss</u> | <u>Validation Loss</u> | F1 Macro | F1 Micro | Precision | Macro | Recall | Macro |
|--------------|----------------------|------------------------|----------|----------|-----------|----------|--------|-------|
| 1 | 0.101800 | 0.088414 | 0.044699 | 0.747292 | 0.044805 | 0.044593 | | |
| 2 | 0.078800 | 0.067347 | 0.045134 | 0.756272 | 0.044817 | 0.045455 | | |
| 3 | 0.054900 | 0.065220 | 0.099639 | 0.758741 | 0.094009 | 0.114037 | | |
| 4 | 0.044300 | 0.052757 | 0.106500 | 0.763066 | 0.133624 | 0.100649 | | |
| 5 | 0.038900 | 0.038783 | 0.430858 | 0.870918 | 0.482591 | 0.411267 | | |
| 6 | 0.031300 | 0.033181 | 0.552250 | 0.900763 | 0.589993 | 0.543019 | | |
| 7 | 0.031500 | 0.028978 | 0.631228 | 0.916793 | 0.704773 | 0.599268 | | |
| 8 | 0.022600 | 0.029914 | 0.635241 | 0.905547 | 0.707039 | 0.609711 | | |
| 9 | 0.013600 | 0.026358 | 0.732029 | 0.915805 | 0.768157 | 0.724414 | | |
| 10 | 0.014400 | 0.024589 | 0.768787 | 0.926471 | 0.800899 | 0.784669 | | |
| 11 | 0.009000 | 0.022660 | 0.843469 | 0.942731 | 0.885630 | 0.829884 | | |
| 12 | 0.007500 | 0.022253 | 0.861222 | 0.944606 | 0.877820 | 0.869707 | | |
| 13 | 0.006900 | 0.022633 | 0.855481 | 0.941691 | 0.881049 | 0.860784 | | |
| 14 | 0.005800 | 0.022659 | 0.860536 | 0.944606 | 0.879966 | 0.863457 | | |
| 15 | 0.005600 | 0.022517 | 0.846470 | 0.940146 | 0.868224 | 0.848685 | | |
| 16 | 0.004900 | 0.022890 | 0.850799 | 0.941520 | 0.875800 | 0.848685 | | |

Figura 5.4.1.5.5 Resultados Definitivos EMO con test

La Figura 5.4.1.5.5 muestra los resultados obtenidos con una partición 80–10–10, donde el modelo alcanza un valor de F1 Macro del 78 % sobre el conjunto de test, lo que indica un comportamiento estable en la evaluación sobre datos no vistos.

| <u>Epoch</u> | <u>Training Loss</u> | <u>Validation Loss</u> | F1 Macro | F1 Micro | Precision | Macro | Recall | Macro |
|--------------|----------------------|------------------------|----------|----------|-----------|----------|--------|-------|
| 1 | 0.794700 | 0.615834 | 0.239655 | 0.413557 | 0.154504 | 0.918343 | | |
| 2 | 0.453500 | 0.429092 | 0.415134 | 0.600729 | 0.295953 | 0.915147 | | |
| 3 | 0.258600 | 0.391643 | 0.662600 | 0.817839 | 0.584204 | 0.889280 | | |
| 4 | 0.178600 | 0.348593 | 0.718573 | 0.867950 | 0.653203 | 0.901407 | | |
| 5 | 0.136900 | 0.444522 | 0.783379 | 0.904628 | 0.748710 | 0.866059 | | |
| 6 | 0.127500 | 0.450563 | 0.801141 | 0.912331 | 0.785034 | 0.848987 | | |
| 7 | 0.041600 | 0.463812 | 0.817837 | 0.927599 | 0.804781 | 0.864128 | | |
| 8 | 0.034100 | 0.545044 | 0.830183 | 0.928623 | 0.827032 | 0.866118 | | |
| 9 | 0.052200 | 0.578534 | 0.804463 | 0.913545 | 0.791967 | 0.845794 | | |
| 10 | 0.025000 | 0.564846 | 0.803915 | 0.907527 | 0.779870 | 0.845620 | | |
| 11 | 0.017900 | 0.603219 | 0.829721 | 0.924188 | 0.831551 | 0.854302 | | |
| 12 | 0.011800 | 0.609596 | 0.812167 | 0.915230 | 0.795453 | 0.847174 | | |
| 13 | 0.010900 | 0.631843 | 0.816814 | 0.919192 | 0.816994 | 0.841454 | | |
| 14 | 0.006900 | 0.647509 | 0.827748 | 0.928986 | 0.837395 | 0.853140 | | |
| 15 | 0.006800 | 0.680905 | 0.829034 | 0.927746 | 0.831749 | 0.846429 | | |
| 16 | 0.005200 | 0.676482 | 0.831761 | 0.929088 | 0.838374 | 0.846429 | | |

Figura 5.4.1.5.6 Resultados Definitivos EMO

Por su parte, la Figura 5.4.1.5.6 presenta los resultados del entrenamiento sin conjunto de test explícito, empleando una partición 80–20. En esta configuración, se obtienen valores de F1 Macro notablemente superiores a los observados en fases previas para esta tarea.

No obstante, estos resultados deben interpretarse con cautela. La mejora observada está fuertemente condicionada por las características del conjunto de datos utilizado, cuya distribución se aproxima a un esquema mayoritariamente unietiqueta. Esta circunstancia simplifica la tarea de clasificación y puede dar lugar a valores de F1 Macro artificialmente elevados en comparación con escenarios multietiqueta más complejos.

Desde la fase inicial de selección del modelo, en la que se obtenía un valor de F1 Macro = 0.5929, hasta la configuración final tras la optimización de hiperparámetros, el modelo alcanza un valor de 0.8318. Si bien esta mejora es cuantitativamente notable, no refleja necesariamente una mejora proporcional en la capacidad general de análisis emocional, sino que está influida por el

tamaño reducido y la simplicidad del conjunto de datos disponible para esta tarea.

| Tarea | Modelo final | Split | F1 Macro inicial | F1 Macro final |
|-------|-----------------|-------|------------------|----------------|
| EP | RoBERTa-base | 80/20 | 0.671 | 0.782 |
| EF | DeBERTa-v3-base | 80/20 | 0.612 | 0.675 |
| EMO | RoBERTa-base | 80/20 | 0.593 | 0.832 |

Figura 5.4.1.5.7 Resumen Resultados finales

2 La Figura 5.4.1.5.7 presenta un resumen de los resultados finales obtenidos en las tres tareas de clasificación consideradas. En todos los casos, los modelos seleccionados muestran mejoras consistentes tras la optimización de hiperparámetros y el ajuste del esquema de entrenamiento

Cabe destacar que los entrenamientos se realizaron sobre conjuntos de datos de tamaño limitado y con un notable desbalanceo entre etiquetas, lo que afecta especialmente a la predicción de las clases minoritarias. Por este motivo, se ha utilizado el F1 Macro como métrica principal de evaluación, ya que proporciona una visión más equilibrada del rendimiento global del sistema.

5.4.2 Encoder-Decoder Transformer

Los modelos Transformer de tipo encoder-decoder combinan un módulo encoder, encargado de generar representaciones semánticas del texto de entrada, con un módulo decoder que utiliza dichas representaciones para generar una secuencia de salida. Este tipo de arquitectura resulta especialmente adecuado para tareas generativas, como traducción automática, resumen o reformulación, aunque también puede adaptarse a tareas de análisis mediante una formulación adecuada del problema.

En el presente trabajo, los modelos encoder-decoder se han evaluado como alternativa a las arquitecturas encoder-only, con el objetivo de analizar si su capacidad generativa aporta ventajas en tareas de clasificación y análisis contextual del texto. Para ello, dichas tareas se han planteado como problemas de generación de secuencias, en los que el modelo produce una salida estructurada que codifica las etiquetas asociadas a cada instancia.

31 Dado que el conjunto de datos fue modificado respecto a versiones previas del sistema, se llevó a cabo una comparativa desde cero entre distintos modelos encoder-decoder, empleando en todos los casos los mismos hiperparámetros básicos estándar con el fin de garantizar la comparabilidad de los resultados. A partir de esta evaluación inicial, se seleccionó el modelo con mejor rendimiento para su análisis posterior.

Los modelos encoder-decoder empleados en los experimentos han sido los siguientes:

- T5 [41]
- facebook/bart-base [42]

- Flan-T5 [43]

Todos los modelos se entrenaron bajo las mismas condiciones experimentales y se evaluaron de forma independiente para las tareas de postura epistémica (EP) y postura efectiva (EF). La categoría emocional (EMO) no se abordó mediante modelos encoder-decoder, ya que se encontraba en una fase experimental temprana y se optó por un enfoque alternativo basado en etiquetado a nivel de token, más adecuado para las características del conjunto de datos disponible.

Antes de presentar los resultados obtenidos y la selección final del modelo encoder-decoder, se describen a continuación las métricas utilizadas para la evaluación del rendimiento:

Epoch:

Es una pasada completa del modelo por todo el conjunto de entrenamiento.

Training Loss:

Mide el error del modelo sobre los datos de entrenamiento, se calcula usando una función de pérdida (*cross-entropy loss* en Transformers) e indica qué tan bien el modelo está aprendiendo los patrones del conjunto de entrenamiento.

Validation Loss:

Mide el error del modelo sobre un conjunto de validación, que no se utiliza para entrenar, sirve para evaluar la capacidad de generalización del modelo, se calcula con la misma función de pérdida que el training loss.

Exact Match [44]:

Es una métrica de evaluación estricta, muy usada en tareas de generación de secuencias, vale 1 si la salida generada por el modelo coincide exactamente con la referencia, vale 0 si hay cualquier diferencia, aunque sea mínima.

5.4.2.1 Comparación de entrenamientos ENTRENAMIENTOS ENCODER-DECODER TRANSFROMER(EP)

En esta sección se analizan los resultados obtenidos mediante modelos Transformer de tipo encoder-decoder en la tarea de etiquetación epistémica (EP). A diferencia de los modelos encoder-only empleados en apartados anteriores, estas arquitecturas permiten la generación explícita de secuencias de salida, lo que resulta especialmente adecuado para tareas de inserción y generación de texto anotado.

Para la evaluación del rendimiento se ha utilizado la métrica **Exact Match** [44], que mide el porcentaje de ejemplos en los que la secuencia generada por el modelo coincide exactamente con la secuencia de referencia.

| Epoch | Training Loss | Validation Loss | Exact Match |
|-------|---------------|-----------------|-------------|
| 1 | 0.106200 | 0.062035 | 0.510172 |
| 2 | 0.045800 | 0.041191 | 0.647887 |
| 3 | 0.034900 | 0.040371 | 0.644757 |
| 4 | 0.024500 | 0.041463 | 0.674491 |
| 5 | 0.018000 | 0.042864 | 0.674491 |
| 6 | 0.010600 | 0.045231 | 0.674491 |
| 7 | 0.004800 | 0.049783 | 0.682316 |
| 8 | 0.010500 | 0.050317 | 0.694836 |
| 9 | 0.006800 | 0.052764 | 0.691706 |
| 10 | 0.005000 | 0.053668 | 0.697966 |

Figura 5.4.2.1.1 Resultados t5 EP para comparar

7 En la Figura 5.4.2.1.1 se muestran los resultados obtenidos por el modelo T5 durante el proceso de entrenamiento. La *training loss* presenta una disminución progresiva, mientras que la *validation loss* se mantiene relativamente estable, lo que indica un proceso de aprendizaje consistente.

En cuanto a la métrica **Exact Match**, el modelo alcanza valores cercanos al 68–69 %, lo que refleja una capacidad razonable para generar correctamente las etiquetas epistémicas en la secuencia de salida, aunque con margen de mejora.

| Epoch | Training Loss | Validation Loss | Exact Match |
|-------|---------------|-----------------|-------------|
| 1 | 0.130200 | 0.072235 | 0.460094 |
| 2 | 0.082900 | 0.074599 | 0.486698 |
| 3 | 0.072200 | 0.047560 | 0.564945 |
| 4 | 0.055800 | 0.055667 | 0.575900 |
| 5 | 0.045700 | 0.056394 | 0.569640 |
| 6 | 0.026600 | 0.051790 | 0.568075 |
| 7 | 0.007600 | 0.055661 | 0.585290 |
| 8 | 0.005700 | 0.079034 | 0.558685 |
| 9 | 0.007800 | 0.074813 | 0.585290 |
| 10 | 0.002000 | 0.074250 | 0.586854 |

Figura 5.4.2.1.2 Resultados facebook/bart-base EP para comparar

23 La Figura 5.4.2.1.2 presenta los resultados correspondientes al modelo facebook/bart-base. Durante el entrenamiento se observa una reducción progresiva de la *training loss*, lo que indica que el modelo es capaz de ajustarse adecuadamente a los datos de entrenamiento.

No obstante, la *validation loss* presenta ligeras oscilaciones y no sigue una tendencia claramente descendente, lo que sugiere una menor estabilidad en la

capacidad de generalización del modelo. En cuanto a la métrica *Exact Match*, se aprecia una mejora inicial que posteriormente se estabiliza, alcanzando valores cercanos 59 % en las últimas épocas. Estos resultados indican que, aunque el modelo logra aprender patrones relevantes, su rendimiento es inferior al obtenido por otros modelos evaluados en esta tarea.

| Epoch | Training Loss | Validation Loss | Exact Match |
|-------|---------------|-----------------|-------------|
| 1 | 0.089600 | 0.051748 | 0.582160 |
| 2 | 0.052100 | 0.041326 | 0.651017 |
| 3 | 0.026100 | 0.041509 | 0.651017 |
| 4 | 0.027500 | 0.044102 | 0.663537 |
| 5 | 0.021000 | 0.042152 | 0.687011 |
| 6 | 0.014200 | 0.044440 | 0.690141 |
| 7 | 0.012400 | 0.048005 | 0.691706 |
| 8 | 0.007900 | 0.055126 | 0.690141 |
| 9 | 0.006400 | 0.058141 | 0.693271 |
| 10 | 0.005700 | 0.058198 | 0.691462 |

Figura 5.4.2.1.3 Resultados google/flan-t5-base EP para comparar

65 En la Figura 5.4.2.1.3 se presentan los resultados obtenidos por el modelo google/flan-t5-base para la tarea EP. La *training loss* disminuye de forma progresiva a lo largo del entrenamiento, mientras que la *validation loss* se mantiene relativamente estable, con ligeras variaciones entre épocas.

68 La métrica *Exact Match* muestra una tendencia ascendente, alcanzando valores cercanos al 69 % en las últimas épocas. Este comportamiento evidencia una mejora respecto a BART, aunque sin superar de forma clara los resultados obtenidos por T5.

A partir del análisis comparativo de los tres modelos evaluados, se selecciona T5 como modelo encoder-decoder de referencia para la tarea epistémica, al alcanzar el mayor valor de *Exact Match*, cercano al 70 %. Este resultado puede considerarse satisfactorio, ya que implica que el modelo es capaz de generar correctamente aproximadamente siete de cada diez secuencias de etiquetas epistémicas de forma exacta.

5.4.2.2 Comparación de entrenamientos ENTRENAMIENTOS ENCODER-DECODER TRANSFROMER(EF)

19 En esta sección se analizan los resultados obtenidos a partir de los entrenamientos realizados para la tarea de etiquetación efectiva (EF) mediante modelos Transformer de tipo encoder-decoder. Para ello, se han evaluado tres arquitecturas en sus versiones base, con el objetivo de identificar el modelo que 5 ofrece el mejor rendimiento en términos de generación correcta de etiquetas.

| Epoch | Training Loss | Validation Loss | Exact Match |
|-------|---------------|-----------------|-------------|
| 1 | 0.271900 | 0.203777 | 0.222378 |
| 2 | 0.159900 | 0.116501 | 0.395804 |
| 3 | 0.109600 | 0.092826 | 0.521678 |
| 4 | 0.079400 | 0.086404 | 0.565035 |
| 5 | 0.081900 | 0.082689 | 0.583217 |
| 6 | 0.068600 | 0.079970 | 0.594406 |
| 7 | 0.062500 | 0.077584 | 0.618182 |
| 8 | 0.053000 | 0.077101 | 0.620979 |
| 9 | 0.051500 | 0.077197 | 0.618182 |
| 10 | 0.062300 | 0.077108 | 0.625175 |

Figura 5.4.2.2.1 Resultados t5 EF para comparar

7 En la Figura 5.4.2.2.1 se muestran los resultados obtenidos por el modelo T5 durante el proceso de entrenamiento en la tarea EF. La *training loss* presenta una disminución progresiva a lo largo de las épocas, lo que indica un aprendizaje adecuado por parte del modelo.

1 La *validation loss* desciende de forma pronunciada en las primeras épocas y posteriormente se estabiliza, lo que sugiere un proceso de entrenamiento consistente, sin indicios claros de sobreajuste. En cuanto a la métrica *Exact Match*, se observa una mejora gradual conforme avanza el entrenamiento, alcanzando valores cercanos al 62 % en la última época, lo que refleja una capacidad moderada del modelo para generar secuencias correctas en esta tarea.

| Epoch | Training Loss | Validation Loss | Exact Match |
|-------|---------------|-----------------|-------------|
| 1 | 0.187600 | 0.116302 | 0.379021 |
| 2 | 0.107200 | 0.078669 | 0.447552 |
| 3 | 0.065900 | 0.079032 | 0.472727 |
| 4 | 0.049800 | 0.071263 | 0.521678 |
| 5 | 0.040500 | 0.071467 | 0.537063 |
| 6 | 0.032000 | 0.073280 | 0.535664 |
| 7 | 0.023400 | 0.077971 | 0.549650 |
| 8 | 0.017400 | 0.079592 | 0.545455 |
| 9 | 0.014400 | 0.085865 | 0.544056 |
| 10 | 0.023400 | 0.085523 | 0.553846 |

Figura 5.4.2.2.2 Resultados facebook/bart-base EF para comparar

23 La Figura 5.4.2.2.2 presenta los resultados correspondientes al modelo facebook/bart-base en la tarea EF. La *training loss* muestra una reducción constante durante el entrenamiento, indicando que el modelo logra ajustarse a los datos de entrenamiento.

13

No obstante, la *validation loss* presenta oscilaciones entre épocas, lo que sugiere una menor estabilidad en términos de generalización. La métrica *Exact Match* aumenta progresivamente en las primeras épocas y tiende a estabilizarse en torno al 55 %, situándose por debajo del rendimiento obtenido por otros modelos evaluados.

| Epoch | Training Loss | Validation Loss | Exact Match |
|-------|---------------|-----------------|-------------|
| 1 | 0.222700 | 0.173355 | 0.248951 |
| 2 | 0.150000 | 0.105068 | 0.468531 |
| 3 | 0.101900 | 0.087304 | 0.549650 |
| 4 | 0.077000 | 0.081462 | 0.594406 |
| 5 | 0.077600 | 0.078175 | 0.622378 |
| 6 | 0.069200 | 0.076028 | 0.627972 |
| 7 | 0.057700 | 0.076632 | 0.633566 |
| 8 | 0.045900 | 0.075875 | 0.633566 |
| 9 | 0.050800 | 0.075901 | 0.634965 |
| 10 | 0.068300 | 0.075257 | 0.634965 |

Figura 5.4.2.2.3 Resultados google/flan-t5-base EF para comparar

7 En la Figura 5.4.2.2.3 se muestran los resultados obtenidos por el modelo google/flan-t5-base para la tarea EF. Al igual que en los modelos anteriores, la *training loss* disminuye de forma progresiva a lo largo del entrenamiento, reflejando un proceso de aprendizaje estable.

1 La *validation loss* se mantiene relativamente constante tras las primeras épocas, lo que indica una buena capacidad de generalización. La métrica *Exact Match* presenta una tendencia creciente, alcanzando valores cercanos al 63 % en las últimas épocas, superando tanto a T5 como a BART en esta tarea.

A partir del análisis comparativo de los resultados, se selecciona google/flan-t5-base como modelo encoder-decoder de referencia para la tarea EF, al presentar un rendimiento ligeramente superior a T5 y claramente mejor que BART, con un valor de *Exact Match* cercano al 63 %.

5.4.2.3 Resultados finales

En el caso de los modelos encoder-decoder no se ha aplicado un proceso de optimización de hiperparámetros mediante Optuna. La principal razón es que, en este tipo de arquitecturas, el objetivo principal no es maximizar métricas de clasificación, sino la generación de texto coherente y su comparación exacta con secuencias de referencia. Aunque una optimización de hiperparámetros podría producir mejoras marginales, estas no serían comparables a las obtenidas en modelos encoder-only.

Además, el elevado coste computacional asociado a los modelos encoder-decoder hace que la aplicación de técnicas de optimización automática resulte poco eficiente. Por ejemplo, el entrenamiento de un único modelo T5 puede requerir aproximadamente siete veces más tiempo que un modelo encoder-only

como RoBERTa. Si a ello se añadiera un proceso de optimización con múltiples *trials*, el tiempo total de ejecución podría prolongarse fácilmente más de un día.

| Modelo | Exact Match final (%) | Comportamiento del entrenamiento | Observaciones |
|---------------------|-----------------------|---|---------------------------------|
| T5 | ≈ 69–70 % | Entrenamiento estable, validation loss controlada | Mejor rendimiento global en EP |
| google/flan-t5-base | ≈ 69 % | Aprendizaje estable, ligera mejora sobre BART | Muy cercano a T5 |
| facebook/bart-base | ≈ 59 % | Validation loss inestable | Rendimiento claramente inferior |

Figura 5.4.2.3.1 Tabla resumen – Modelos Encoder–Decoder Transformer (EP)

Tabla resumen – Modelos Encoder–Decoder Transformer (EF)

| Modelo | Exact Match final (%) | Comportamiento del entrenamiento | Observaciones |
|---------------------|-----------------------|--|-----------------------|
| T5 | ≈ 62 % | Entrenamiento consistente, mejora progresiva | Buen rendimiento base |
| google/flan-t5-base | ≈ 63 % | Estable y mejor generalización | Mejor modelo en EF |
| facebook/bart-base | ≈ 55 % | Oscilaciones en validation loss | Rendimiento inferior |

Figura 5.4.2.3.2 Tabla resumen – Modelos Encoder–Decoder Transformer (EF)

En las Figuras 5.4.2.3.1 y 5.4.2.3.2 se presentan las tablas resumen correspondientes a los modelos encoder–decoder evaluados en las tareas EP y EF, respectivamente.

En la dimensión epistémica (EP), el modelo seleccionado ha sido T5, alcanzando un valor de *Exact Match* cercano al 70 %, lo que indica una elevada capacidad para generar correctamente las etiquetas epistémicas. En la dimensión efectiva (EF), el modelo seleccionado ha sido google/flan-t5-base, que presenta el mejor rendimiento global, con valores próximos al 63 % de *Exact Match*.

Al igual que en los modelos encoder-only, una vez finalizado el entrenamiento definitivo, se genera una carpeta de resultados que contiene los archivos producidos durante el proceso, los cuales se suben posteriormente a la plataforma Hugging Face para su almacenamiento y reutilización.

5.4.3 Entrenamiento BIO

Con el objetivo de abordar las limitaciones observadas en la tarea de etiquetación emocional (EMO), se desarrolló un sistema alternativo de entrenamiento basado en el esquema de etiquetado **BIO (Begin–Inside–Outside)**. Este enfoque se aplica a la detección de expresiones emocionales a nivel de token y constituye una línea de trabajo complementaria a los métodos empleados en las categorías epistémica y efectiva.

Este sistema se encuentra todavía en una fase experimental, ya que fue implementado en un periodo de tiempo limitado y se encuentra en proceso de evolución. Por este motivo, los resultados obtenidos deben interpretarse como **preliminares**, y su finalidad principal es evaluar la viabilidad del enfoque y su potencial de mejora frente a los métodos anteriores.

9 El sistema utiliza un modelo **encoder-only RoBERTa**, formulando el problema como una tarea de clasificación secuencial a nivel de token, en la que cada palabra de la secuencia recibe una etiqueta **B**, **I** u **O** en función de su pertenencia a una expresión emocional. Este planteamiento resulta especialmente adecuado, dado que los modelos encoder-only están optimizados para tareas de etiquetado secuencial y presentan un menor coste computacional en comparación con arquitecturas encoder-decoder.

Asimismo, el uso del esquema BIO permite emplear métricas de evaluación más informativas, como **precisión**, **recall** y **F1**, proporcionando una visión más detallada del comportamiento del modelo que métricas estrictas como *Exact Match*, especialmente en escenarios con anotaciones parciales o solapadas.

Para la implementación de este enfoque fue necesario adaptar el proceso de generación y anotación de los conjuntos de datos. En concreto, se modificaron los scripts encargados del procesamiento (`process_emo.py`) con el fin de producir anotaciones compatibles con el esquema BIO.

El sistema desarrollado extrae automáticamente los *spans* relevantes a partir de textos anotados con marcas del tipo `<<EMO_...>>`, transformándolos en etiquetas BIO a nivel de palabra. Posteriormente, los textos se tokenizan y las etiquetas se alinean con los subtokens generados por el modelo RoBERTa. El entrenamiento se realiza mediante un clasificador a nivel de token implementado con **AutoModelForTokenClassification[]**, y se evalúa utilizando la métrica Entity-F1 proporcionada por la librería `seqeval[]`. El mejor modelo se selecciona en función de dicha métrica.

Finalmente, el modelo entrenado se emplea en la fase de inferencia para predecir etiquetas BIO sobre nuevos textos, reconstruyendo las anotaciones `<<EMO_...>>` mediante un proceso de post-procesado.

70 5.5 HUGGING FACE

5 Hugging Face es una plataforma de código abierto especializada en el desarrollo y despliegue de modelos de lenguaje basados en arquitecturas Transformer. Proporciona un ecosistema completo de librerías, modelos preentrenados y herramientas que facilitan tanto el entrenamiento como la adaptación y la puesta en producción de modelos de *deep learning* de forma eficiente.

Una vez finalizado el proceso de entrenamiento, los artefactos generados se almacenan y despliegan en Hugging Face. En el contexto de este trabajo, se han creado y publicado distintos modelos, cargando los pesos obtenidos durante el entrenamiento y exponiéndolos a través de Hugging Face Spaces, que actúan como entornos de ejecución accesibles desde la web. Cada Space se apoya en una aplicación definida en el archivo `app.py`, responsable de gestionar la lógica de inferencia y la interacción con el usuario.

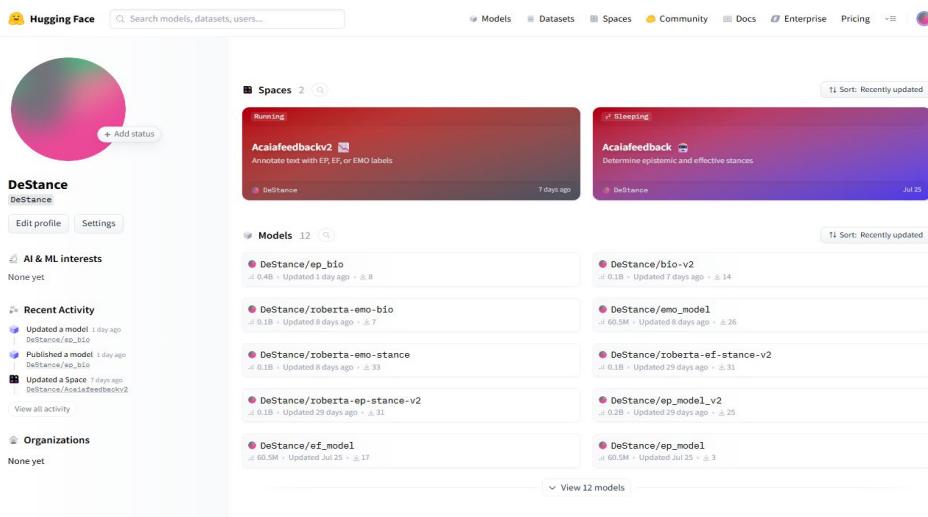


Figura 5.5.1 Hugging Face – DeStance

La Figura 5.5.1 muestra la página principal del repositorio del proyecto en Hugging Face, donde se pueden observar los distintos Spaces creados. En concreto, se dispone del Space **Acaiafeedback**, correspondiente a una versión inicial del sistema, y del Space **Acaiafeedbackv2**, que representa una versión posterior con mejoras funcionales. Asimismo, se listan los distintos modelos entrenados y publicados, incluyendo versiones preliminares y modelos destinados a pruebas o futuras extensiones, lo que permite mantener un registro completo de la evolución del sistema.

5.5.1 App.py

El archivo *app.py* constituye el núcleo funcional de los Spaces de Hugging Face, ya que define la lógica de inferencia, la integración entre los distintos modelos y la interfaz de usuario. Gracias a este archivo, los modelos entrenados pueden transformarse en aplicaciones accesibles y ejecutables en la nube sin necesidad de infraestructuras adicionales.

En una primera fase, el Space **Acaiafeedback** contaba con una implementación básica de *app.py*, que permitía clasificar un texto utilizando un único modelo seleccionado mediante un menú desplegable. En esta versión, cada ejecución devolvía una etiqueta junto con su puntuación de confianza en el caso de los modelos RoBERTa, mientras que los modelos T5 generaban únicamente el texto anotado, sin asociar puntuaciones explícitas. Esta implementación inicial tenía como objetivo validar el funcionamiento básico de los modelos entrenados y comprobar su correcta integración en la plataforma.

Tras el análisis de las limitaciones detectadas, se desarrolló una nueva versión del sistema, **Acaiafeedbackv2**, con una aplicación *app.py* considerablemente más avanzada. En esta versión, el texto de entrada se segmenta en frases con el fin de mejorar la precisión del análisis. Los modelos RoBERTa se emplean para realizar clasificación multietiqueta y calcular puntuaciones de confianza, mientras que los modelos T5 se utilizan únicamente cuando existe evidencia suficiente para generar anotaciones textuales. Asimismo, se

incorporan umbrales configurables que permiten controlar la sensibilidad del sistema y ajustar su comportamiento a distintos escenarios de uso.

De forma adicional, se integra un módulo específico de análisis emocional (EMO) basado en un enfoque de etiquetado BIO/NER, que permite insertar etiquetas directamente en el texto sin fragmentar palabras. La aplicación se apoya en una interfaz interactiva desarrollada con Gradio, lo que facilita la interacción con el usuario, el ajuste dinámico de parámetros y la interpretación de los resultados obtenidos.

5.5.1.2 Evolución del sistema

Con el fin de ilustrar la evolución del sistema, se utiliza como ejemplo un texto previamente anotado con etiquetas epistémicas y afectivas. El texto mostrado a continuación representa el resultado esperado tras un proceso de anotación correcto, incluyendo distintos casos de posicionamiento discursivo:

17 And I'm afraid <EP_CGA_P> it is naive and misguided to think that only men can be covert operatives or that refugee flows would <EP_EM> not be subject to some form of exploitation.
4 Sadly, the human cost of this crisis will <EP, EM> worsen as Putin continues his indiscriminate shelling. This is already leading to the displacement of people into eastern Europe. The UN Refugee Agency said this morning that 660,000 Ukrainian refugees have fled since the invasion last Thursday. I know <EP, FV, P> the Government announced today that the UK's Ukrainian refugee scheme will now be extended to wider family members. These Benches have been calling repeatedly on the Government to do more to help. There will <EP, EM> be considerable relief that they have now changed their position and accepted that we must <EF, DM, P> do more.

19 Con el objetivo de evaluar la capacidad de detección automática del sistema, las etiquetas originales se eliminan manualmente y el texto limpio se introduce en las distintas versiones de la aplicación.

The screenshot shows the Acaiafeedback application interface. On the left, there is a text input area labeled "Texto a clasificar" containing the cleaned-up text from the previous step. Below this is a "Modelo" dropdown set to "roberta-ep-stance". At the bottom are "Clear" and "Submit" buttons. On the right, there is a "output" panel displaying JSON code: { "label": "EP_EM", "score": 0.992 }. Below the output panel is a "Share via Link" button.

Figura 5.5.1.2 .1 Sistema de Acaiafeedback roberta-ep

15 La Figura 5.5.1.2.1 muestra el funcionamiento de la versión inicial del sistema Acaiafeedback, tal y como fue entregado originalmente, utilizando un modelo RoBERTa entrenado para la detección de posicionamiento epistémico.

Text to clasificar

```

1   {
2     "label": "EP_CGA_P"
3     "score": "N/A"
4   }
  
```

Modelo

ep_model

Clear Submit Share via Link

Figura 5.5.1.2 .2 Sistema de Acaiafeedback ep-model

En la Figura 5.5.1.2.2 se presenta la ejecución correspondiente al modelo T5 en la misma versión inicial del sistema.

En esta primera implementación, los modelos RoBERTa detectan únicamente una parte de las etiquetas esperadas y presentan dificultades para procesar textos extensos de forma coherente. Por su parte, los modelos T5 generan anotaciones incompletas o incorrectas, siendo incapaces de identificar la totalidad de las etiquetas presentes en el texto. Estos resultados reflejan las limitaciones funcionales de la versión inicial del sistema.

15

Anotador (EP / EF / EMO) con T5 + RoBERTa (EMO usa BIO)

Texto limpio

And I'm afraid it is naive and misguided to think that only men can be covert operatives or that refugee flows would not be subject to some form of exploitation.

Sadly, the human cost of this crisis will worsen as Putin continues his indiscriminate shelling. This is already leading to the displacement of people into eastern Europe. The UN Refugee Agency said this morning that 660,000 Ukrainian refugees have fled since the invasion last Thursday. I know the Government announced today that the UK's Ukrainian refugee scheme will now be extended to wider family members. These Benches have been calling repeatedly on the Government to do more to help. There will be considerable relief that they have now changed their position and accepted that we must do more.

Umbral mínimo de score (RoBERTa)

0.9 0 1 32 280 512

Longitud máxima de generación (T5)

Tipo de anotación / modelo

EP EF EMO

Texto anotado

Etiquetas ([MODO][S] label score)

- [EP][S1] EP_CGA_P 0.999
- [EP][S2] EP_EM 0.988
- [EP][S2] EP_CGA 0.999
- [EP][S2] EP_CGA 0.002
- [EP][S3] NO_EP 0.999
- [EP][S3] EP_CGA 0.999
- [EP][S4] EP_CGA 0.999
- [EP][S4] EP_CGA_P 0.000
- [EP][S5] EP_PV_P 1.000
- [EP][S5] EP_EM 0.019
- [EP][S6] NO_EP 0.999
- [EP][S6] EP_IE 0.000
- [EP][S7] EP_EM 0.999
- [EP][S7] EP_CGA 0.003

Anotar y clasificar

Usar vía API Construido con Gradio Configuración

Figura 5.5.1.2 .3 Sistema de Acaiafeedbackv2

La Figura 5.5.1.2.3 muestra el resultado obtenido con la versión mejorada del sistema, denominada Acaiafeedbackv2. En esta versión se observa una interpretación más coherente y completa de los textos analizados.

Una de las principales mejoras introducidas en Acaiafeedbackv2 es la corrección de los problemas relacionados con la longitud del texto procesable por los modelos RoBERTa y T5. En la versión anterior, la segmentación del texto en

frases se realizaba desde el front-end de la plataforma DeStance, lo que generaba inconsistencias durante la ejecución del análisis en el sistema.

En Acaiafeedbackv2, la segmentación del texto se lleva a cabo directamente en el propio sistema, dentro del archivo *app.py*. Esta modificación permite un control más preciso del flujo de procesamiento y evita desajustes entre componentes. Además, se introduce un límite configurable de longitud máxima por frase, ajustable mediante el parámetro de longitud de generación, lo que mejora la estabilidad y el rendimiento del análisis, especialmente en los modelos T5.

Adicionalmente, la interfaz permite ajustar dinámicamente parámetros clave del análisis, como el umbral mínimo de confianza de los modelos RoBERTa y la longitud máxima de generación de los modelos T5. La separación visual entre el texto de entrada y el texto anotado refuerza la trazabilidad del proceso, permitiendo comparar de forma directa la entrada original con el resultado generado.

El sistema es ahora capaz de detectar correctamente casos de clasificación multietiqueta, como se observa en la primera sentencia del ejemplo, donde se identifican simultáneamente varias categorías epistémicas sin pérdida de precisión. Para cada sentencia, se muestra la etiqueta detectada junto con su puntuación de confianza. Las sentencias sin posicionamiento epistémico se clasifican como *NO_EP* con alta fiabilidad.

The screenshot shows the Acaiafeedbackv2 EMO annotation interface. At the top, it says "Anotador (EP / EF / EMO) con T5 + RoBERTa (EMO usa BIO)".
The "Texto limpio" section contains the text: "The lack of translation services felt insecure <>EMO_SEC_NEG<> to new arrivals. Workers felt resentful <>EMO_SAT_NEG<> of the unfair treatment. The council's handling of the complaint seemed correct <>EMO_MOR_MIX<>, but its spirit remained open to interpretation."
Below this are configuration sliders: "Umbral mínimo de score (RoBERTa)" set to 0.5, "Longitud máxima de generación (T5)" set to 32, and a "Tipo de anotación / modelo" section with radio buttons for EP, EF, and EMO (which is selected).
The "Texto anotado" section shows the same text with additional annotations: "[EMO][S1] EMO_SEC_NEG 0.981", "[EMO][S1] EMO_QUA_NEG 0.005", "[EMO][S1] EMO_SAT_NEG 0.017", "[EMO][S1] EMO_MOR_NEG 0.004", "[EMO][S1] EMO_MOR_MIX 0.930", and "[EMO][S1] EMO_MOR_PDS 0.011".
At the bottom, there is a "Anotar y clasificar" button and links for "Usar vía API", "Configuración", and "Configuración".

Figura 5.5.1.2 .4 Sistema de Acaiafeedbackv2 EMO

La Figura 5.5.1.2.4 presenta la ejecución del módulo de análisis emocional (EMO) en la versión Acaiafeedbackv2. Este módulo permite anotar expresiones emocionales directamente en el texto mediante un modelo basado en el esquema BIO, posibilitando la identificación precisa de fragmentos emocionales a nivel de token. Esta funcionalidad no estaba disponible en versiones anteriores del sistema ACAIA-FEEDBACK.

Finalmente, la interfaz ofrece un acceso directo a la funcionalidad de uso mediante API, lo que permite integrar el sistema en plataformas externas. Este

mecanismo es el que se emplea para la integración del sistema de análisis en la plataforma DeStance, aspecto que se describe en el apartado correspondiente.

5.5.2 Importación a DeStance

Una vez desplegado el *Space* en Hugging Face y verificado el correcto funcionamiento de los modelos, el siguiente paso consiste en integrar dicho servicio con el *front-end* de la plataforma DeStance, de modo que los modelos puedan ser consumidos desde la aplicación web y sus resultados presentados en el entorno final de uso.

Hugging Face proporciona distintos mecanismos de acceso a los modelos desplegados a través de cada *Space*, disponibles mediante una sección específica de uso vía API.

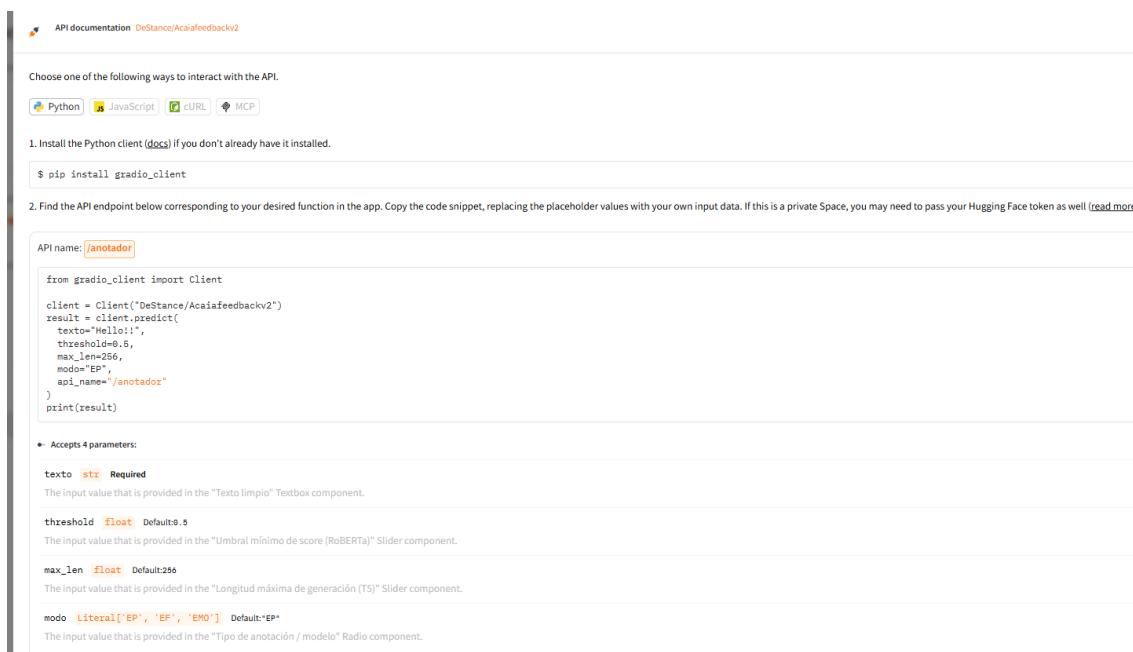


Figura 5.5.2.1 Modos API

En la Figura 5.5.2.1 se muestran las distintas opciones y guías proporcionadas por Hugging Face para la integración de los modelos, incluyendo ejemplos de consumo mediante diferentes lenguajes y herramientas, como Python, JavaScript y cURL. Estas opciones facilitan la integración del sistema en aplicaciones externas de forma flexible.

Dado que la conexión con los modelos se realiza desde el *front-end* de DeStance, se optó por utilizar la opción basada en JavaScript. Esta integración se apoya en el *endpoint* asociado al *Space* correspondiente y en el envío de peticiones HTTP que contienen el texto a analizar, recibiendo como respuesta las anotaciones generadas por los modelos.

La comunicación con Hugging Face se implementó en un archivo específico del *front-end*, denominado *moocService.ts*, encargado de centralizar las llamadas a

la API. En este archivo se define la función responsable de enviar el texto al *endpoint* del *Space* y de procesar la respuesta devuelta por el modelo, permitiendo que el resto de la aplicación consuma los resultados del análisis de forma transparente y desacoplada.

```
// Cliente de Hugging Face

let hfClient: any = null;

async function getClient() {
    if (!hfClient) {
        hfClient = await Client.connect("DeStance/Acaiafeedbackv2");
    }
    return hfClient;
}
```

Figura 5.5.2.2 Conexión con el bak-end

En la Figura 5.5.2.2 muestra simplemente como hacemos la conexión en nuestro back-end, para conectar a nuestro space de Hugging Face

5.6 MOOCs

Una vez entrenados y optimizados los modelos, desplegados los *Spaces* de Hugging Face y habilitada la comunicación entre estos y el *front-end* de la aplicación, el último paso del desarrollo consiste en integrar los resultados del análisis en la plataforma web De-Stance y visualizarlos de forma accesible para el usuario final.

La plataforma De-Stance se encuentra actualmente en desarrollo en el marco de distintos trabajos académicos, y constituye el entorno en el que se integra el módulo ACAIA-FEEDBACK. En la página principal de la aplicación se presenta un acceso al módulo MOOC, que corresponde a la parte de la plataforma relacionada con el análisis discursivo y que concentra las funcionalidades desarrolladas en este Trabajo de Fin de Grado.

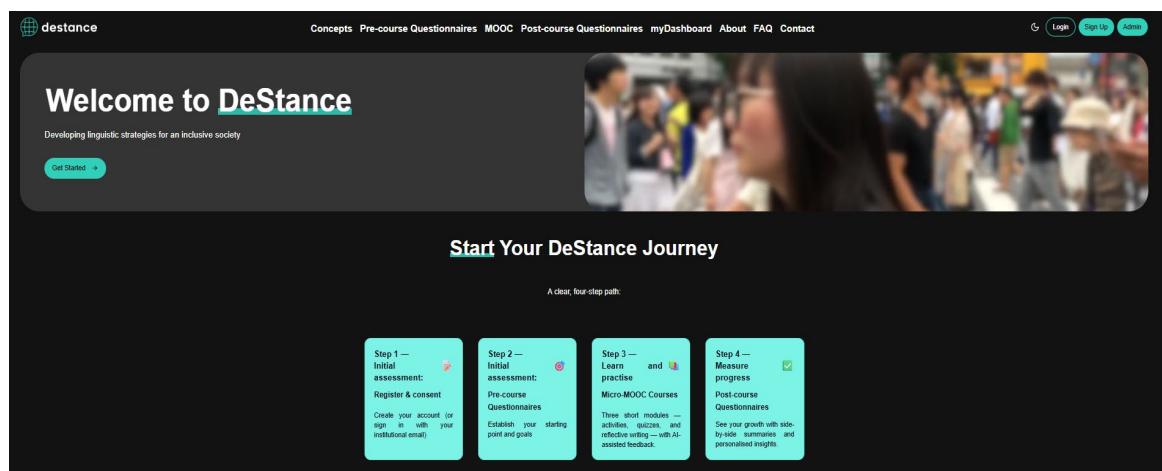


Figura 5.6.1 Pagina inicio DeStance

La Figura 5.6.1 muestra la página principal de la plataforma DeStance, donde se observa la estructura general de la aplicación y el acceso a los distintos módulos, incluyendo el módulo MOOC.



Figura 5.6.2 Sección MOOC DeStance

En la Figura 5.6.2 se presenta la sección MOOC de la plataforma. Desde esta vista, el usuario puede acceder a las distintas sesiones temáticas disponibles, cada una orientada al análisis de una dimensión concreta del discurso. El módulo MOOC incluye las siguientes sesiones:

- Session 1 – Epistemic Stance
- Session 2 – Emotional Stance
- Session 3 – Effective Stance
- Session 4 – Stance & Source Check

Las sesiones 1, 2 y 3 presentan una estructura y funcionalidad equivalente, diferenciándose únicamente en el tipo de etiquetas analizadas (epistémicas, emocionales o afectivas). La Session 4 corresponde a un módulo independiente desarrollado en otro Trabajo de Fin de Grado realizado por mi compañero Juan Locos.

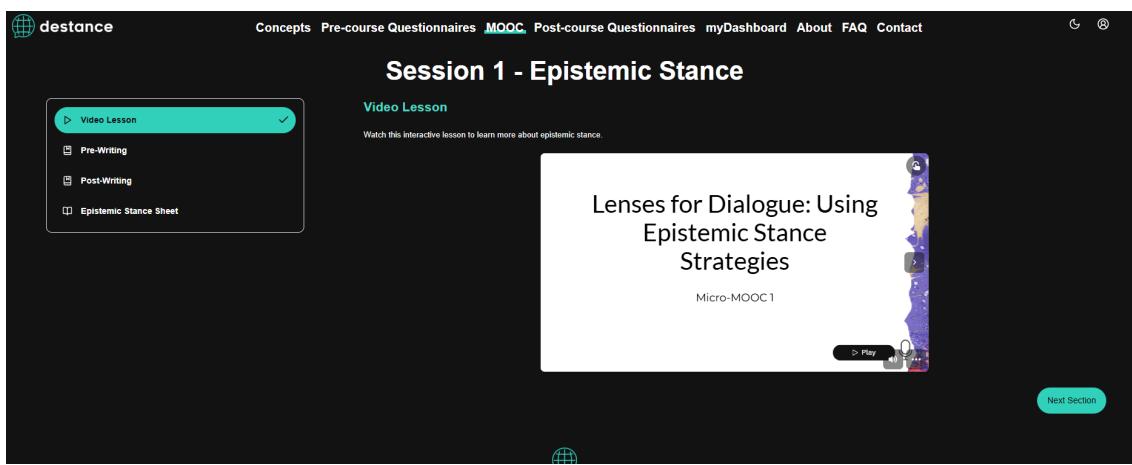


Figura 5.6.3 MOOC Epistemic Stance Video Lesson

En la Figura 5.6.3 se muestra la vista correspondiente a la Session 1 – Epistemic Stance, concretamente la sección inicial de tipo *Video Lesson*. En esta fase, el estudiante recibe una introducción teórica al concepto de *epistemic stance*, proporcionando el contexto necesario para comprender el análisis que se

realizará posteriormente. Esta parte tiene un carácter formativo y sirve como preparación para las fases prácticas del módulo.

Cuando se termina el video se pulsa Next Section y pasamos a la parte del Pre Writing

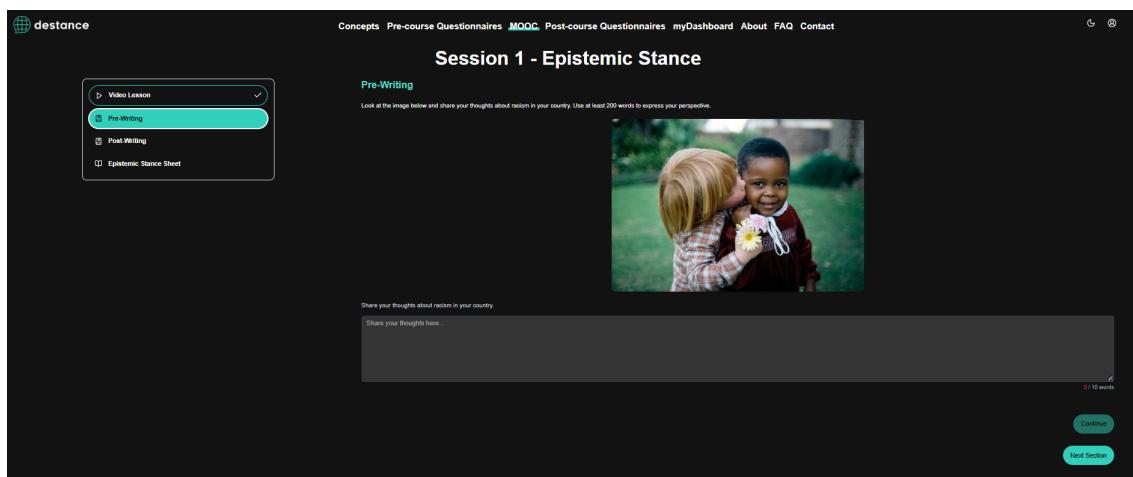


Figura 5.6.4 MOOC Epistemic Stance Pre-Writing

La Figura 5.6.4 ilustra la fase de *Pre-Writing* dentro de la sesión Epistemic Stance. En esta etapa, se solicita al estudiante la redacción de un texto a partir de una consigna concreta (por ejemplo, “*Share your thoughts about racism in your country*”). Se recomienda que el texto esté redactado en inglés y presente una puntuación adecuada, ya que el análisis se realiza a nivel de frase.

Una vez introducido el texto, el usuario puede iniciar el proceso de análisis mediante la opción *Analyze*, lo que desencadena la ejecución del modelo correspondiente a través de la conexión con Hugging Face. El sistema procesa automáticamente el texto y genera las etiquetas asociadas a cada fragmento, integrando los modelos desarrollados en este trabajo dentro de un flujo interactivo orientado al usuario final.

Session 1 - Epistemic Stance

Pre-Writing

Look at the image below and share your thoughts about racism in your country. Use at least 200 words to express your perspective.



Share your thoughts about racism in your country.

Once they arrive in Greece, they are kept in detention for up to three to four months, before being released on the border between Greece and Turkey. They end up in Athens, but their destination of choice is the UK and western Europe.

Illegal migration is the No. 1 issue facing the EU, and although it is recognised as such in the Commission's programme, under the heading "Towards a New Policy on Migration", actually we do not hear enough from the Commission and Ministers about this critical issue. It requires a new deal with the countries of north Africa, particularly in respect of how the Egyptians, Libyans, Algerians, and to a lesser extent the Moroccans and Tunisians, deal with the people traffickers, who take up to €10,000 each from each migrant on the boat and then leave them, sometimes without a captain, in the hope that the Italian Government will EP_EM send ships to save them, which does not always happen. So although it is not necessarily EP_EM on the conscience of people sitting in this House, it is certainly EP_EM on the conscience of the Commission if it has one. A €5 million programme is being established between FRONTEX—the agency of the EU that manages co-operation between national border guards to secure the external borders of the Union, including from illegal immigration, human trafficking and terrorist infiltration—and the UNHCR to help the countries of the western Balkans strengthen their asylum and migration policies and capabilities. Additionally, FRONTEX is co-operating closely with Turkey, which has helped to stem the flow of the ghost ships that we saw EP_IIE_P before Christmas. I understand that EP_CGA_P it has assigned a member state expert to help the Turks to improve security around the port of Mersin. The EU has also signed a readmission agreement with Turkey, which means EP_IIR that Turkey must take back not only nationals who may EP_EM be irregular migrants, but migrants who are seen EP_JIE to have come from Turkey. There is close and joint co-operation in the Aegean Sea and on the Greek and Bulgarian land borders. Could the Minister explain how on earth the UK could begin to influence or support these actions if we were outside the EU?

394 / 10 words

Analysis Results

Analyzed text

We know EP_FV_P what pressure the Greeks are under, because of their economic situation, and people are flooding into Turkey from Iraq and Syria, despite the efforts of the Turkish Government. Once they arrive in Greece, they are kept in detention for up to three to four months, before being released on the border between Greece and Turkey. They end up in Athens, but their destination of choice is the UK and western Europe. Illegal migration is the No. 1 issue facing the EU, and although it is recognised as such in the Commission's programme, under the heading "Towards a New Policy on Migration", actually EP_FV we do not hear enough from the Commission and Ministers about this critical issue. It requires a new deal with the countries of north Africa, particularly in respect of how the Egyptians, Libyans, Algerians, and to a lesser extent the Moroccans and Tunisians, deal with the people traffickers, who take up to €10,000 each from each migrant on the boat and then leave them, sometimes without a captain, in the hope that the Italian Government will EP_EM send ships to save them, which does not always happen. So although it is not necessarily EP_EM on the conscience of people sitting in this House, it is certainly EP_EM on the conscience of the Commission if it has one. A €5 million programme is being established between FRONTEX—the agency of the EU that manages co-operation between national border guards to secure the external borders of the Union, including from illegal immigration, human trafficking and terrorist infiltration—and the UNHCR to help the countries of the western Balkans strengthen their asylum and migration policies and capabilities. Additionally, FRONTEX is co-operating closely with Turkey, which has helped to stem the flow of the ghost ships that we saw EP_IIE_P before Christmas. I understand that EP_CGA_P it has assigned a member state expert to help the Turks to improve security around the port of Mersin. The EU has also signed a readmission agreement with Turkey, which means EP_IIR that Turkey must take back not only nationals who may EP_EM be irregular migrants, but migrants who are seen EP_JIE to have come from Turkey. There is close and joint co-operation in the Aegean Sea and on the Greek and Bulgarian land borders. Could the Minister explain how on earth the UK could begin to influence or support these actions if we were outside the EU?

Labels (label - score)

- [EP][S1] EP_FV_P - 1.000
- [EP][S5] EP_FV - 0.999
- [EP][S6] EP_EM - 1.000
- [EP][S7] EP_EM - 0.999
- [EP][S9] EP_IIE_P - 1.000
- [EP][S10] EP_CGA_P - 0.999
- [EP][S11] EP_IIR - 1.000
- [EP][S11] EP_IIE - 0.996
- [EP][S11] EP_EM - 0.986

Marker type

| Marker type | Number of occurrences |
|-------------|-----------------------|
| EP_FV | 3 |
| EP_FV_P | 2 |
| EP_EM | 2 |
| EP_CGA_P | 1 |
| EP_IIE | 1 |
| EP_IIR | 1 |
| EP_IIE_P | 1 |
| EP_APH_P | 1 |
| EP_APH | 1 |
| EP_MEP_P | 1 |

Total labels 9

Facts (EP_FV, EP_FV_P) 44.4%
Inferences (EP_IIE, EP_IIE_P, EP_IIR) 22.2%
Modality (EP_EM, EP_CGA_P, EP_CGA, EP_APH_P, EP_APH) 33.3%

Feedback

Regularly incorporating Epistemic Modals into your writing or speaking, which are expressions indicating degrees of probability or certainty, showcases your skill in conveying the level of confidence in your statements. By integrating these modals effectively, you can enhance the credibility and persuasiveness of your writing or speaking. "perhaps", "maybe", "possibly", "probably", "certainly", "surely", "in all probability", "in all likelihood", "must be", "one could", "you could", "everyone could", "everybody could", "he could", "she could", "it could", "that could", "this could", "these could", "those could", "they could", "may", "might", "possible", "likely", "probable", "sure", "certain", "it is clear", "it is evident", "it is obvious", "it's clear", "it's evident", "it's obvious", "clearly", "obviously", "apparently", "seemingly", "it seems that", "it appears", "it feels that", "it seems as if", "it feels as if", "to judge by", "judging by"

Figura 5.6.5 MOOC Epistemic Stance Resultado Pre- Writing

33 En la Figura 5.6.5 se muestra el resultado obtenido tras la ejecución del análisis en la fase de *Pre-Writing*. El texto introducido se presenta de forma anotada, mostrando las etiquetas generadas por el modelo para cada fragmento. Debajo del texto anotado se incluye un panel de etiquetas con las puntuaciones de confianza asociadas, proporcionando una visión clara del comportamiento del modelo.

63 El análisis se realiza a nivel de frase, y el sistema identifica automáticamente los distintos tipos de etiquetas epistémicas presentes en el texto. Adicionalmente, se han incorporado representaciones gráficas que facilitan la interpretación de los resultados, como gráficos de tipo *donut* y gráficos de líneas, que muestran la distribución y frecuencia relativa de las etiquetas detectadas.

3 Los resultados textuales y estadísticos se almacenan en una base de datos MongoDB, lo que permite su reutilización y análisis posterior.

En cuanto al aspecto visual, los cambios generados por el análisis se reflejan mediante distintos elementos gráficos. En particular, se utiliza un gráfico donut para representar la distribución de las etiquetas detectadas, y un gráfico de líneas para mostrar la evolución o frecuencia relativa de cada categoría. Estos gráficos se generan automáticamente a partir de los resultados obtenidos y se actualizan dinámicamente en la interfaz.

En versiones iniciales del sistema, los resultados se enviaban directamente desde el *Space* de Hugging Face al *front-end* de la aplicación. No obstante, en la versión actual se ha invertido este flujo: el *front-end* envía el texto completo al *Space* de Hugging Face, donde se ejecuta el análisis, y los resultados se devuelven posteriormente a la aplicación. Este cambio tiene como objetivo principal mejorar la flexibilidad y el mantenimiento del sistema, ya que permite realizar ajustes en los criterios de segmentación o en los modelos utilizados sin necesidad de desplegar nuevamente la aplicación web. Gracias a esta conexión directa entre el *front-end* y Hugging Face, las actualizaciones se reflejan de forma inmediata.

A screenshot of a web application titled "Session 1 - Epistemic Stance". The top navigation bar includes links for "Concepts", "Pre-course Questionnaires", "MOOC", "Post-course Questionnaires", "myDashboard", "About", "FAQ", and "Contact". On the left, there's a sidebar with dropdown menus for "Video Lesson", "Pre-Writing" (which is currently selected), "Post-Writing", and "Epistemic Stance Sheet". The main content area is titled "Session 1 - Epistemic Stance" and contains a section for "Post-Writing". It asks the user how they feel about racism in their country and provides a text input field for sharing thoughts. A progress bar at the bottom indicates "0 / 10 words". At the bottom right are "Continue" and "Next Section" buttons.

Figura 5.6.6 MOOC Epistemic Stance Pos-Writting

5 La Figura 5.6.6 muestra la fase de *Post-Writing* dentro de la sesión Epistemic Stance. Esta fase presenta una estructura similar a la de *Pre-Writing*, con la diferencia de que el estudiante redacta el texto en respuesta a una consigna distinta. El proceso de análisis es idéntico, lo que permite comparar el desempeño discursivo del usuario antes y después del proceso formativo.

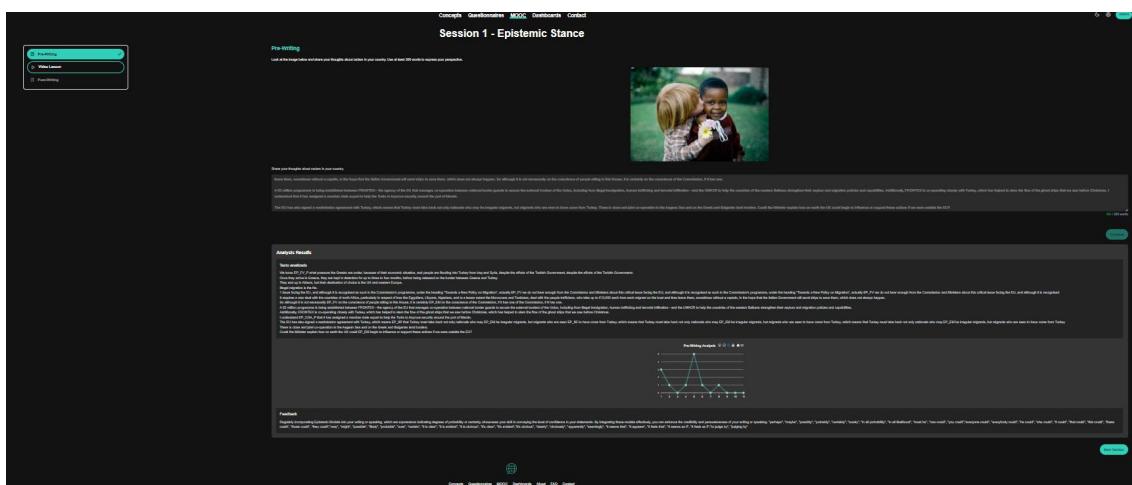


Figura 5.6.7 Estado inicial del sistema

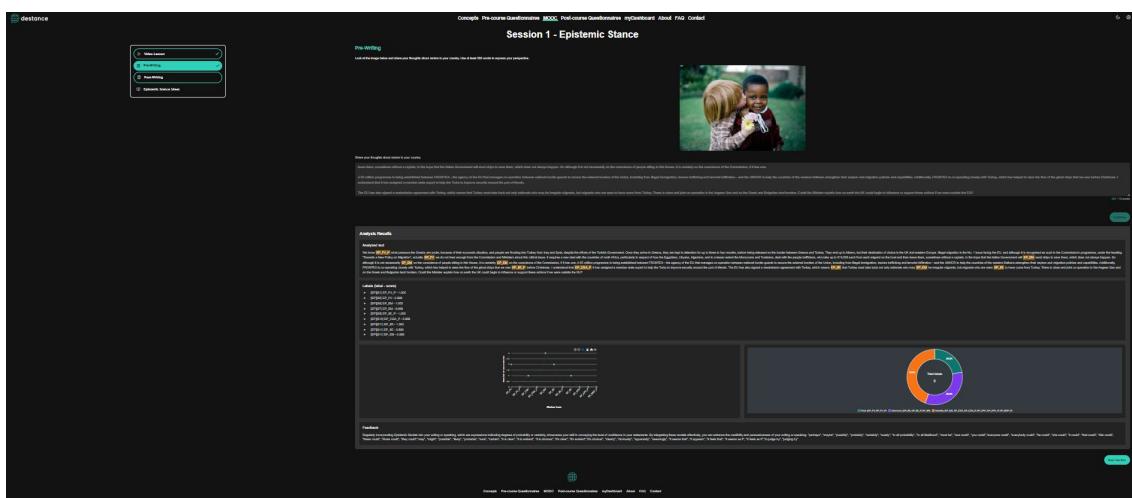


Figura 5.6.8 Estado final del sistema

67 13 1 Finalmente, las Figuras 5.6.7 y 5.6.8 muestran la evolución del sistema desde su estado inicial hasta la versión final implementada en este trabajo. En ellas se reflejan las mejoras alcanzadas tanto en la detección de etiquetas como en el rendimiento de los modelos y en la presentación visual de los resultados.

La integración completa de los modelos de análisis en la plataforma DeStance permite ofrecer un entorno funcional, interactivo y orientado al usuario final, cerrando el ciclo completo desde el entrenamiento de modelos hasta su aplicación práctica en un contexto educativo real. Esta implementación sienta las bases para futuras extensiones y mejoras del módulo MOOC.

6 Resultados y conclusiones

1 En este capítulo resumiremos el impacto y conclusiones de este proyecto, así como los problemas encontrados durante su desarrollo y las soluciones planteadas.

6.1 Resultados finales

Como se ha expuesto a lo largo de esta memoria, el primer paso del trabajo consistió en analizar y comprender las principales limitaciones del estado inicial del proyecto. En dicho estado se detectaron problemas relevantes en la detección de etiquetas mediante modelos encoder-only, en particular con RoBERTa, que únicamente era capaz de identificar la presencia de una etiqueta por texto, independientemente de su número de apariciones. Asimismo, se observaron inconsistencias en las puntuaciones de confianza (*scores*) asociadas a las etiquetas detectadas.

35 Para abordar estas limitaciones, se llevó a cabo un proceso de revisión, tratamiento y unificación de los conjuntos de datos, reorganizando los scripts de generación y creando nuevos *datasets* adaptados a un enfoque multilabel. Este proceso permitió un entrenamiento más coherente con los objetivos del proyecto y una mejora sustancial en el comportamiento de los modelos.

1 En cuanto a los entrenamientos, se emplearon dos enfoques diferenciados: modelos encoder-only y modelos encoder-decoder, cada uno con sus arquitecturas correspondientes. Ambos enfoques alcanzaron resultados satisfactorios, con valores de rendimiento superiores al 70 %. Aunque estas métricas son inferiores a las obtenidas en versiones anteriores del sistema, esta disminución está plenamente justificada. En esta nueva fase, los entrenamientos se realizaron sobre conjuntos de datos multilabel y se incorporaron ejemplos negativos (frases sin etiquetas), con el objetivo de mejorar la capacidad de generalización del sistema, lo que impacta directamente en la precisión global.

Los modelos encoder-only obtuvieron resultados ligeramente superiores al 70 %, mientras que los modelos encoder-decoder se situaron en torno a ese mismo valor. Estos resultados pueden considerarse positivos teniendo en cuenta la complejidad añadida del problema. En el contexto de la API y de la aplicación web, ambos tipos de modelos se evalúan de forma independiente, lo que puede dar lugar a pequeñas discrepancias entre las etiquetas detectadas por cada uno. No obstante, este comportamiento no representa una limitación significativa y puede mitigarse mediante la incorporación progresiva de nuevos ejemplos al conjunto de entrenamiento.

Por otro lado, la integración con Hugging Face ha experimentado una de las mejoras más notables del proyecto, tanto en la interpretación de los textos de entrada como en la presentación de los resultados. Estas mejoras se reflejan tanto a nivel académico como visual, dando lugar a una aplicación más eficiente, interpretable e interactiva.

Finalmente, en lo relativo a la aplicación web y a los resultados mostrados al usuario, se observa una evolución coherente con la experimentada en la

plataforma de Hugging Face, con mejoras globales en rendimiento, claridad y experiencia de uso.

6.2 Escalabilidad y mantenibilidad

La nueva implementación desarrollada en este trabajo ha supuesto una mejora significativa en la escalabilidad y mantenibilidad del sistema. En versiones anteriores, el proceso de detección y segmentación de frases se realizaba directamente en el *front-end* de la plataforma, lo que implicaba que cualquier modificación en la lógica de análisis requería retirar el despliegue de la aplicación y volver a desplegarla, dificultando la evolución del sistema.

Con la arquitectura actual, este proceso se ha trasladado íntegramente a Hugging Face, de modo que la detección y separación de frases se realiza directamente en los *Spaces* donde se ejecutan los modelos. Esta decisión permite que cualquier ajuste en los criterios de segmentación, en los modelos empleados o en su configuración se refleje de forma inmediata a través de la API, sin necesidad de modificar ni redesplegar la plataforma DeStance.

Asimismo, esta nueva arquitectura facilita la creación, prueba y sustitución de nuevos modelos de forma ágil, permitiendo experimentar con distintas arquitecturas o versiones sin interrumpir el funcionamiento de la aplicación web. De este modo, el sistema adquiere una mayor flexibilidad, escalabilidad y capacidad de evolución, sentando las bases para futuras mejoras y ampliaciones del proyecto.

6.3 Contribución del proyecto

El proyecto RACISMMAFF ha constituido la base sobre la que se han desarrollado tanto la plataforma DeStance como el módulo ACAIA-FEEDBACK. En este contexto, el presente Trabajo de Fin de Grado contribuye de forma directa al Objetivo de Desarrollo Sostenible número 16 (ODS 16) de la Agenda 2030, orientado a la promoción de sociedades más justas, pacíficas e inclusivas [46].

Mediante el sistema desarrollado, se proporciona una herramienta tecnológica que favorece el análisis y la reflexión crítica sobre el discurso, permitiendo identificar posicionamientos epistémicos, efectivos y emocionales en textos escritos. Este tipo de análisis contribuye a la detección de patrones discursivos relacionados con la desinformación, el racismo o la polarización, alineándose con los principios del ODS 16 en materia de acceso a la información, fortalecimiento de instituciones inclusivas y promoción del pensamiento crítico.

Asimismo, el proyecto presenta una contribución indirecta al Objetivo de Desarrollo Sostenible número 4 (ODS 4), relativo a la educación de calidad [47]. Gracias a la integración del módulo ACAIA-FEEDBACK en la plataforma DeStance, los estudiantes de la asignatura *English for Professional and Academic Communication* pueden recibir feedback automático e inmediato sobre

1
19

12

75

54

1

36 sus textos escritos en inglés, tanto en los cuestionarios previos como posteriores disponibles en la plataforma.

Esta funcionalidad no solo mejora la experiencia de aprendizaje del alumnado, fomentando la autorreflexión y la mejora progresiva de la competencia escrita, sino que también contribuye a reducir la carga de trabajo del profesorado, que anteriormente debía realizar la clasificación y anotación de los textos de forma manual. De este modo, el sistema desarrollado apoya prácticas educativas más sostenibles, escalables y accesibles, en línea con los objetivos planteados por la Agenda 2030.

6.4 Desarrollos futuros del proyecto

De cara a trabajos futuros, se identifican diversas líneas de mejora y ampliación del sistema desarrollado que permitirían incrementar tanto su rendimiento como su robustez y aplicabilidad.

34 En primer lugar, resulta fundamental la obtención de un mayor número de ejemplos para las categorías EP, EF y EMO, así como la incorporación de frases críticas detectadas durante el uso continuado de la plataforma. Este incremento y refinamiento de los conjuntos de datos permitiría enriquecer el entrenamiento de los modelos y mejorar su precisión y capacidad de generalización.

Actualmente, los modelos encoder-only empleados abordan la tarea desde un enfoque multilabel a nivel de sentencia, lo que presenta la limitación de no distinguir múltiples ocurrencias de una misma etiqueta dentro de una oración. Para solventar esta limitación, se considera especialmente prometedora la extensión del esquema de etiquetado BIO, ya aplicado con éxito en el análisis EMO, también a las categorías EP y EF.

La adopción de un sistema BIO unificado permitiría una integración más estrecha entre los componentes encoder y decoder, simplificando la arquitectura actual y mejorando la coherencia global del sistema. Asimismo, resultaría de interés consolidar completamente la categoría EMO, mejorando su rendimiento y completando su integración en la aplicación web.

Por otro lado, sería relevante explorar modelos de mayor tamaño o de última generación, como RoBERTa-large o DeBERTa-large, así como modelos multilingües como XLM-RoBERTa, con el objetivo de ampliar la plataforma DeStance a otros idiomas. Estas líneas de trabajo estarían condicionadas por la disponibilidad de recursos computacionales adecuados, como GPUs o TPUs.

Finalmente, en relación con la aplicación web, una posible mejora consistiría en ampliar la funcionalidad posterior al *post-writing* de los MOOCs, incorporando una sección comparativa que permita analizar de forma más detallada la evolución discursiva del usuario entre las distintas fases.

6.5 Conclusión

En conjunto, los resultados obtenidos confirman que los objetivos definidos en el Capítulo 1 han sido alcanzados, demostrando la viabilidad del sistema para el análisis automático de posicionamientos epistémicos, efectivos y emocionales en contextos educativos reales.

29 Desde el punto de vista técnico, el desarrollo del proyecto ha permitido consolidar conocimientos en el ámbito del aprendizaje automático, así como en el uso de librerías y herramientas para el entrenamiento, evaluación y optimización de modelos de procesamiento del lenguaje natural. Asimismo, se ha profundizado en el uso del lenguaje Python y en el desarrollo de aplicaciones web mediante JavaScript y Svelte, integrando de forma efectiva componentes de *machine learning* y *front-end*.

107 En definitiva, este Trabajo de Fin de Grado ha supuesto un proceso de aprendizaje exigente y altamente enriquecedor, permitiendo aplicar conocimientos teóricos en un entorno práctico y reforzando competencias técnicas y metodológicas relevantes para el ámbito de la ingeniería informática.

6.6 Agradecimientos

1 Para finalizar esta memoria, se desea expresar un sincero agradecimiento a las tutoras Jelena Bobkina Lukaszuka y Elena Domínguez Romero por su orientación, apoyo y disponibilidad a lo largo de todo el desarrollo del proyecto.

81 Asimismo, se agradece la contribución de los compañeros que participaron en fases previas del proyecto, en especial a Ana Clara Murillo García por el trabajo desarrollado anteriormente, y a Juan Locos por su colaboración en el desarrollo de la plataforma web DeStance.

Bibliografía

- 16 90
- [1] Ministerio de Inclusión, Seguridad Social y Migraciones, "Preocupante aumento del racismo y la xenofobia: la FRA advierte sobre una Europa cada vez menos inclusiva," 2025. [Online]. Available:
<https://www.inclusion.gob.es/web/oberaxe/w/preocupante-aumento-del-racismo-y-la-xenofobia-la-fra-advierte-sobre-una-europa-cada-vez-menos-inclusiva>
- 16 87
- [2] Universidad Complutense de Madrid, "Stance strategies in immigration and racism related," s.f. [Online]. Available:
<https://www.ucm.es/racismmaff/about-the-project>
- 1
- [3] DeStance, "Plataforma DeStance," 2026. [Online]. Available:
<https://www.destance.com/>
- 25 1
- [4] A. C. Murillo García, "Análisis automático de textos en De-Stance," B.S. thesis, ETSI Informáticos, Universidad Politécnica de Madrid, Madrid, Spain, 2025. [Online]. Available:
https://oa.upm.es/91053/1/TFG_ANA_CLARA_MURILLO_GARCIA.pdf
- 25 1
- [5] G. Álvarez García, "Trabajo Fin de Grado sobre la plataforma De-Stance," B.S. thesis, ETSI Informáticos, Universidad Politécnica de Madrid, Madrid, Spain, 2025. [Online]. Available:
https://oa.upm.es/90162/1/TFG_GONZALO_ALVAREZ_GARCIA.pdf
- 25 106
- [6] D. Ramón Robertson, "Módulos interactivos en De-Stance," B.S. thesis, ETSI Informáticos, Universidad Politécnica de Madrid, Madrid, Spain, 2025. [Online]. Available:
https://oa.upm.es/89550/3/TFG_Daniel_Ramon_Robertson_1.pdf
- 1
- [7] Grammarly. (2025), "Página oficial 'All-in-one writing assistant for desktop and browser'". [Online]. Available:
<https://www.grammarly.com/>
- [8] ProWritingAid. (2025), "Bring your story to life". [Online]. Available:
<https://plataformasinc.es/quillbot/>
- [9] QuillBot. (2025), "Write better everywhere". [Online]. Available:
<https://quillbot.com/>

29
11

6

1

1

9

62

53

8

18

18

6

38

18

73

1

18

18

- [10] J. Ranalli, T. Neri, and R. Wette, "Automated written corrective feedback: How well can students make use of it?" 2022. [Online]. Available: <https://eric.ed.gov/?id=EJ1198748>
- [11] OpenAI. (2024), "GPT-4 Technical Report". [Online]. Available: https://cdn.openai.com/papers/gpt-4.pdf?itid=lk_inline_enhanced-template
- [12] VOYANT. (2025), "see through your text". [Online]. Available: <https://voyant-tools.org/>.
- [13] Hugging Face, "The AI community building the future," s.f. [Online]. Available: <https://huggingface.co/>
- [14] R. Leow and R. Manchón, "The effects of written corrective feedback processing conditions on L2 learners' written texts and levels of depth of processing," 2021. [Online]. Available: <https://dialnet.unirioja.es/servlet/tesis?codigo=314972>
- [15] T. Wambsganss et al., "Enhancing argumentative writing with automated feedback and social comparison nudging," 2022. [Online]. Available: <https://www.researchgate.net/publication/364133342>
- [16] Hugging Face, "Transformers documentation," 2025. [Online]. Available: <https://huggingface.co/docs/transformers/es/index>
- [17] PyTorch, "PyTorch documentation," 2025. [Online]. Available: <https://pytorch.org/>
- [18] TensorFlow, "TensorFlow documentation," 2025. [Online]. Available: <https://www.tensorflow.org/>
- [19] JAX, "JAX documentation," 2025. [Online]. Available: <https://jax.readthedocs.io/>
- [20] Hugging Face, "Transformers: Auto Classes," Version 3.0.2, 2019. [Online]. Available: https://huggingface.co/transformers/v3.0.2/model_doc/auto.html
- [21] Hugging Face, "Hugging Face Spaces documentation," 2025. [Online]. Available: <https://huggingface.co/docs/hub/spaces>
- [22] Optuna, "Optuna documentation," 2025. [Online]. Available: <https://optuna.org/>
- [23] Google Research, "Google Colaboratory," 2025. [Online]. Available: <https://colab.research.google.com/>
- [24] Mozilla, "JavaScript documentation," 2025. [Online]. Available: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [25] Svelte, "Svelte documentation," 2025. [Online]. Available: <https://svelte.dev/docs>
- [26] MongoDB, "MongoDB documentation," 2025. [Online]. Available: <https://www.mongodb.com/docs/>

- 1 [27] Python Software Foundation, "Python documentation," 2025. [Online]. Available: <https://docs.python.org/3/>
- 8 [28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019. [Online]. Available: <https://aclanthology.org/N19-1423/>
- 21 [29] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," arXiv:1910.10683, 2020. [Online]. Available: <https://arxiv.org/pdf/1910.10683>
- 40 [30] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," arXiv:2005.11401, 2021. [Online]. Available: <https://arxiv.org/pdf/2005.11401>
- 41 [31] Chakki-works, "Seqeval: A Python framework for sequence labeling evaluation," 2023. [Online]. Available: <https://github.com/chakki-works/seqeval>
- [32] M. Carretero and E. Domínguez, "Epistemic words and expressions," Proyecto RACISMMAFF, Universidad Politécnica de Madrid, documento interno, in press.
- [33] M. Carretero and E. Domínguez, "Effective words and expressions," Proyecto RACISMMAFF, Universidad Politécnica de Madrid, documento interno, in press.
- [34] M. Carretero and E. Domínguez, "Emotional words and expressions," Proyecto RACISMMAFF, Universidad Politécnica de Madrid, documento interno, in press.
- 27 [35] J. R. Martin and P. R. R. White, The Language of Evaluation: Appraisal in English. London, UK: Palgrave Macmillan, 2005.
- 30 [36] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," arXiv:2006.03654, 2021. [Online]. Available: <https://arxiv.org/pdf/2006.03654>
- 24 [37] K. Clark et al., "ELECTRA: Pre-training text encoders as discriminators rather than generators," arXiv:2003.10555, 2020. [Online]. Available: <https://arxiv.org/pdf/2003.10555>
- 28 [38] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," arXiv:1907.11692, 2019. [Online]. Available: <https://arxiv.org/pdf/1907.11692>
- 6 [39] Scikit-learn, "Metrics and scoring: Quantifying the quality of predictions," 2009. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html

51
42

22

22

38

8

32

79
6

6

- [40] A. Masoud, "Understanding micro, macro, and weighted averages for scikit-learn metrics," 2022. [Online]. Available: <https://iamirmasoud.com/2022/06/19/understanding-micro-macro-and-weighted-averages>
- [41] Hugging Face, "T5 model documentation," 2025. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/t5
- [42] Hugging Face, "Flan-T5 model documentation," 2025. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/flan-t5
- [43] Hugging Face, "facebook/bart-base model card," 2025. [Online]. Available: <https://huggingface.co/facebook/bart-base>
- [44] Ragas, "Traditional NLP evaluation metrics: Exact match," s.f. [Online]. Available: https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/traditional/
- [45] Criteria for annotating epistemic and effective stance: A proposal for discourse on irregular immigration and humanitarian crises involving refugees.
- [46] Naciones Unidas, "Objetivo 16: Paz, justicia e instituciones sólidas," Agenda 2030 para el Desarrollo Sostenible, 2015. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/peace-justice/>
- [47] Naciones Unidas, "Objetivo 4: Educación de calidad," Agenda 2030 para el Desarrollo Sostenible, 2015. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/education/>

ANEXO

ANEXO A

Repositorio de código de procesado y entrenamientos

El código fuente desarrollado en el marco del presente Trabajo de Fin de Grado, correspondiente a los módulos de procesado de datos, **entrenamiento y evaluación de los modelos** descritos en este trabajo, se encuentra disponible en un repositorio público alojado en la plataforma GitHub.

El repositorio incluye los scripts necesarios para la reproducción de los experimentos presentados y el análisis de los resultados obtenidos.

Repositorio:

<https://github.com/Guille94899/TFG>

El código correspondiente al desarrollo de la plataforma web DeStance (front-end y back-end) no se encuentra disponible públicamente, al formar parte de un proyecto institucional. No obstante, dicho código se encuentra disponible para su revisión junto con el presente Trabajo de Fin de Grado, si el tribunal lo estima necesario.

ANEXO B

Acceso y funcionamiento de los modelos en Hugging Face

Este anexo describe el acceso público a los modelos y espacios desarrollados en el marco de este Trabajo de Fin de Grado a través de la plataforma Hugging Face, así como la evolución de las herramientas disponibles.

B.1. Acceso a la plataforma

El acceso a los recursos se realiza a través de la página web oficial de Hugging Face. Los modelos y espacios asociados al proyecto se encuentran publicados de forma pública, por lo que no es necesario disponer de una cuenta ni realizar ningún proceso de autenticación para su consulta.

Para localizar los recursos, basta con utilizar la barra de búsqueda de la plataforma e introducir el nombre del usuario “DeStance”. Al seleccionar dicho perfil, se accede a la página que agrupa todos los modelos y espacios creados.

B.2. Spaces disponibles

En el perfil del usuario DeStance se encuentran disponibles dos Hugging Face Spaces para la interacción con los modelos desarrollados:

- ****Acaiafeedback****: corresponde a una versión inicial de la herramienta, desarrollada con anterioridad al presente Trabajo de Fin de Grado. Este Space permite analizar textos y obtener información sobre la postura epistémica y afectiva.
- ****Acaiafeedback v2****: corresponde a la versión actualizada y mejorada, desarrollada en el marco de este Trabajo de Fin de Grado. Esta versión amplía las capacidades del sistema, incorporando mejoras en los modelos utilizados, en el esquema de anotación y en la presentación de los resultados al usuario.

Ambos Spaces pueden ejecutarse directamente desde el navegador mediante una interfaz web interactiva.

B.3. Modelos publicados

Además de los Spaces, el perfil del usuario DeStance incluye múltiples modelos entrenados y publicados, correspondientes a las distintas dimensiones del análisis (postura epistémica, postura efectiva y postura emocional), así como a las diferentes versiones y configuraciones evaluadas durante el desarrollo del proyecto.

B.4. Activación de los Spaces y visualización de archivos

Los Hugging Face Spaces pueden encontrarse en estado “Sleeping” cuando no han sido utilizados recientemente. Este comportamiento es habitual en la plataforma y tiene como objetivo optimizar el uso de recursos computacionales. En caso de que un Space se encuentre en dicho estado, el usuario puede activarlo manualmente haciendo clic en la opción “Restart this Space”. Una vez finalizado este proceso, el Space pasa al estado “Running”, lo que permite interactuar con la aplicación de forma normal.

Cuando el Space se encuentra en ejecución, es posible introducir textos para su análisis y acceder a los archivos asociados al Space a través de la sección Files, incluyendo el código de la aplicación y los ficheros de configuración correspondientes.

El acceso a los modelos y Spaces se realiza desde:

<https://huggingface.co/DeStance>

ANEXO C

Funcionamiento de la plataforma web DeStance

Este anexo describe el funcionamiento de la plataforma web DeStance, desarrollada como entorno principal de interacción con el usuario en el marco del presente Trabajo de Fin de Grado, así como el flujo de uso de los módulos relacionados con el análisis de postura discursiva.

C.1. Acceso y registro de usuarios

64

El acceso a la plataforma se realiza a través de la dirección web oficial:
<https://www.destance.com/>

55

Para utilizar la plataforma es necesario completar un proceso de registro. En primer lugar, el usuario debe seleccionar la opción “Sign up” e introducir una dirección de correo electrónico y una contraseña. A continuación, el sistema solicita información adicional de carácter básico, como el nombre completo, la fecha de nacimiento y el género.

69

Tras completar estos campos, el usuario debe indicar su lengua materna (*mother tongue*), así como información relacionada con el aprendizaje de idiomas, incluyendo el idioma que desea aprender, su nivel de competencia y su nacionalidad. Una vez introducidos estos datos, el sistema envía un código de verificación al correo electrónico proporcionado, que debe ser introducido para completar el proceso de registro.

Finalizado este proceso, el usuario queda dado de alta y puede acceder a las funcionalidades de la plataforma.

C.2. Acceso al módulo MOOC

Una vez autenticado, el usuario puede interactuar con los distintos módulos disponibles en DeStance. En el contexto de este Trabajo de Fin de Grado, el módulo relevante es el módulo MOOC.

Al seleccionar la opción “MOOC”, el sistema muestra las distintas categorías de análisis de postura disponibles. En el momento de redacción de este trabajo, el módulo de postura emocional (*Emotional Stance*) se encuentra en fase de pruebas y no está aún disponible para su uso público.

El usuario puede seleccionar, por ejemplo, la opción “Epistemic Stance” para interactuar con los contenidos y herramientas asociadas a dicha dimensión.

C.3. Flujo de trabajo en Epistemic Stance

Una vez dentro del módulo de *Epistemic Stance*, el usuario accede en primer lugar a un vídeo introductorio (*lesson*) que presenta los conceptos básicos relacionados con este tipo de postura discursiva.

Tras visualizar el contenido introductorio, el usuario puede avanzar a la siguiente sección, correspondiente a la fase de *pre-writing*. En esta fase, el usuario debe introducir un texto en lengua inglesa. El sistema exige un mínimo de diez palabras; en caso contrario, se muestra un mensaje de aviso indicando que el texto introducido es insuficiente.

Una vez introducido un texto válido, el usuario selecciona la opción “Continue”, lo que desencadena la ejecución del análisis automático. El sistema procesa el texto y muestra los resultados correspondientes al análisis de postura epistémica.

C.4. Post-writing y finalización del módulo

Tras revisar los resultados del *pre-writing*, el usuario puede avanzar a la fase de *post-writing*. En esta sección, el usuario introduce un nuevo texto, siguiendo el mismo procedimiento descrito en la fase anterior. El sistema vuelve a ejecutar el análisis y presenta los resultados obtenidos.

Finalmente, el usuario accede a la última sección del módulo, denominada “Epistemic Stance Sheet”, donde se presenta una lección en formato PDF que resume los contenidos trabajados. Una vez revisado este material, el usuario puede completar el módulo seleccionando la opción “Complete course”.

3

5

ANEXO D

Informe de similitud mediante Turnitin

Con el fin de garantizar la originalidad del presente Trabajo de Fin de Grado, el documento completo ha sido sometido a un análisis de similitud utilizando la herramienta Turnitin, ampliamente empleada en el ámbito académico para la detección de coincidencias textuales.

El análisis se realizó una vez finalizada la redacción del documento y tras la incorporación de todas las referencias bibliográficas correspondientes. El informe obtenido refleja un **índice de similitud del 12 %**, valor que se encuentra dentro de los márgenes considerados aceptables para trabajos académicos de carácter técnico y continuista.

Las coincidencias detectadas corresponden principalmente a:

- Referencias bibliográficas y citas correctamente atribuidas.
- Terminología técnica estándar propia del ámbito del procesamiento del lenguaje natural.
- Descripciones metodológicas comunes a trabajos previos desarrollados en el marco de la plataforma DeStance y del proyecto RACISMAFF, todos ellos debidamente citados.

El informe completo de similitud se encuentra disponible en el siguiente enlace proporcionado por la plataforma Turnitin:

Enlace al informe de Turnitin:

https://ev.turnitin.com/app/carta/es/?o=2857279314&student_user=1&s=1&u=115657833&ro=103&lang=es

El acceso al informe puede requerir credenciales institucionales. El informe en formato PDF se encuentra disponible para su consulta por parte del tribunal si fuese necesario. Y también se ha subido el pdf resultado a un contenedor de Git Hub

<https://github.com/Guille94899/TFG/blob/main/MEMORIA%20turniting.pdf>