Universidad de Jaén

Escuela Politécnica Superior de Jaén

DETECCIÓN DE FIABILIDAD EN NOTICIAS EN ESPAÑOL

Autor: Elena Alonso Tejederas

Grado: Ingeniería Informática

Director: Dª. Salud María Jiménez Zafra
D. Miguel Ángel García Cumbreras

Departamento del director: Informática

Fecha: 11/09/2024

Licencia CC

CREA

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Dª Salud María Jiménez Zafra y D. Miguel Ángel García Cumbreras, tutores del Trabajo de Fin de Grado titulado: Detección de fiabilidad en noticias en español, que presenta Elena Alonso Tejederas, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2024

El alumno:

ALONSO TEJEDERAS ELENA - 46271553S
Firmado digitalmente por ALONSO TEJEDERAS ELENA - 46271553S
Fecha: 2024.09.11 22:29:38 +02'00'

Elena Alonso Tejederas

Los tutores:

JIMENEZ ZAFRA SALUD MARIA - 77370897R
Firmado digitalmente por JIMENEZ ZAFRA SALUD MARIA - 77370897R
Fecha: 2024.09.11 20:57:36 +02'00'

Salud María Jiménez Zafra

GARCIA CUMBRERAS MIGUEL ANGEL - 52871272E
Firmado digitalmente por GARCIA CUMBRERAS MIGUEL ANGEL - 52871272E
Fecha: 2024.09.11 20:42:41 +02'00'

Miguel Ángel García Cumbreras

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN....7
1.1. Motivación....8
1.2. Propósito....9
1.3. Objetivos....10
1.4. Resultados esperados....11
1.5. Planificación....11
1.6. Presupuesto....13
1.7. Estructura del proyecto....16
2. ANTECEDENTES....18
2.1. Procesamiento del Lenguaje Natural (PLN)....19
2.2. Técnicas para el preprocesamiento de un texto....20
2.3. Ingeniería de prompt....22
2.3.1. Definición, utilidad y función del prompt....23
2.3.2. Elementos principales....23
2.3.3. Técnicas....25
2.3.3.1. Prompt sin entrenamiento previo (Zero-shot)....25
2.3.3.2. Prompt con pocas muestras (Few-shot)....26
2.3.3.3. Prompt cadena de pensamiento (CoT)....28
2.3.4. Aplicaciones y tipos de prompt....29
2.3.5. Importancia del prompt....33
2.3.6. Aplicación en el proyecto....33
2.3. Fiabilidad del lenguaje en noticias....36
2.3.1. “Fake news”....38
2.3.2. Guía RUN-AS....39
2.4. Campaña de evaluación IberLEF....43
2.5. Tarea FLARES: Clasificación de confiabilidad....44
2.5.1. Corpus....45
2.5.2. Medidas de evaluación....48
3. DESARROLLO....50
3.1. Análisis de datos y métricas de evaluación....51
3.1.1. Análisis del conjunto de datos....51
3.1.2. Análisis de las métricas de evaluación....55
3.2. Sistema de clasificación para la detección de fiabilidad....60
3.3. Técnicas utilizadas de ingeniería de prompt....73
3.3.1. Prompt sin entrenamiento previo (Zero-shot)....73
3.3.2. Prompt con pocas muestras (Few-shot)....75
3.3.3. Prompt cadena de pensamiento (CoT)....78
3.4. Hiperparámetros de configuración....81
3.5. Modelos generativos utilizados....83
3.5.1. Gemini....83
3.5.2. ChatGPT....85
3.6. Post-procesamiento....87

3.7. Tecnologías utilizadas....88
4. RESULTADOS Y DISCUSIÓN....90
4.1. Análisis de resultados....91
4.1.1. Resultados obtenidos con el archivo de prueba....91
4.1.1.1. Resultados obtenidos con la técnica “zero-shot”....92
4.1.1.2. Resultados obtenidos con la técnica “few-shot”....93
4.1.1.3. Resultados obtenidos con la técnica “CoT-zero-shot”....95
4.1.1.4. Resultados obtenidos con la técnica “CoT-few-shot”....96
4.1.2. Resultados obtenidos con el archivo de test....97
4.1.2.1. Resultados obtenidos con la técnica “zero-shot”....98
4.1.2.2. Resultados obtenidos con la técnica “few-shot”....100
4.1.2.3. Resultados obtenidos con la técnica “CoT-zero-shot”....101
4.1.2.4. Resultados obtenidos con la técnica “CoT-few-shot”....103
4.1.3. Conclusiones finales sobre la realización de las pruebas....104
4.2. Análisis de errores....105
4.2.1. Matriz de confusión....106
4.2.2. Detección de patrones erróneos en la clasificación....109
4.3. Comparación de resultados obtenidos por los demás participantes en la Tarea FLARES....111
4.4. Análisis de errores de ejecución....113
5. CONCLUSIONES....116
5.1. Conclusiones sobre resultados finales....116
5.2. Posibles mejoras....118
5.3. Trabajos futuros....120
5.4. Satisfacción personal....121
6. ANEXOS....122
6.1. Anexo A: Descripción de prompts....122
6.1.1. Técnica “zero-shot”....122
6.1.2. Técnica “few-shot”....132
6.1.3. Técnica “CoT-zero-shot”....142
6.1.4. Técnica “CoT-few-shot”....152
6.2. Anexo B: Manuales....163
6.2.1. Manual de instalación....163
6.2.2. Manual de usuario....165
6.2.2.1. Descripción de la carpeta principal....165
6.2.2.2. Instrucciones de uso....167
7. BIBLIOGRAFÍA....175

ÍNDICE DE ILUSTRACIONES

Ilustración 1.1 Diagrama de Gantt de la planificación del proyecto..... 13
Ilustración 2.1 Ejemplo de mejora de rendimiento utilizando cadena de pensamiento..... 28
Ilustración 2.2 RUN-AS Guideline.....40
Ilustración 2.3 Distribución de etiquetas de fiabilidad en archivos.....47
Ilustración 3.1 Distribución de etiquetas 5W1H en relación con el contenido de Tag_Text del conjunto de datos de entrenamiento.....54
Ilustración 3.2 Distribución de etiquetas 5W1H en relación con el contenido de Tag_Text del conjunto de datos de prueba..... 55
Ilustración 3.3 Diagrama de actividad para el flujo de ejecución de cada técnica.....62
Ilustración 3.4 Distribución de etiquetas utilizadas con Gemini y con ChatGPT..... 86
Ilustración 3.5 Logos de las tecnologías utilizadas.....90
Ilustración 4.1 Matriz de confusión..... 107
Ilustración 4.2 Tabla de posiciones de los participantes en la tarea FLARES..... 112
Ilustración 4.3 Primer error de ejecución..... 114
Ilustración 4.4 Segundo error de ejecución.....115
Ilustración 4.5 Tercer error de ejecución..... 115
Ilustración 6.1 Instalación de Jupyter Notebook..... 163
Ilustración 6.2 Instalación de Visual Studio Code.....164
Ilustración 6.3 Carpeta principal vista desde el entorno local.....165
Ilustración 6.4 Contenido de la carpeta “archivos_evaluacion_csv”.....165
Ilustración 6.5 Contenido de la carpeta “conjunto_datos”.....166
Ilustración 6.6 Contenido de la carpeta “evaluacion”.....166
Ilustración 6.7 Contenido de la carpeta “modelos”..... 167
Ilustración 6.8 Contenido de la carpeta “modelos”..... 167
Ilustración 6.9 Carpeta principal vista desde el entorno Jupyter Notebook.....168
Ilustración 6.10 Contenido de la carpeta de un modelo de test.....168
Ilustración 6.11 Primera ruta a modificar (archivo de entrada)..... 169
Ilustración 6.12 Segunda ruta a modificar (archivo de evaluación)..... 169
Ilustración 6.13 Tercera ruta a modificar (archivo de salida)..... 169
Ilustración 6.14 Clave de modelo Gemini a modificar.....169
Ilustración 6.15 Clave de modelo OpenAI a modificar.....170
Ilustración 6.16 Cómo ejecutar todas las celdas en el entorno Jupyter Notebook..... 170
Ilustración 6.17 Resultados de los valores predichos en el entorno Jupyter Notebook.....171
Ilustración 6.18 Entorno de Jupyter Notebook.....171
Ilustración 6.19 Estados del Kernel en el entorno Jupyter Notebook..... 171
Ilustración 6.20 Rutas a modificar según la ubicación de los archivos.....172
Ilustración 6.21 Resultado del archivo “post_procesamiento” por pantalla..... 172
Ilustración 6.22 Resultado de las métricas de evaluación del archivo ejecutado.....173
Ilustración 6.23 Rutas a modificar según la ubicación de los archivos.....173
Ilustración 6.24 Resultados de la matriz de confusión.....174

ÍNDICE DE TABLAS

Tabla 1.1 Planificación del proyecto....12
Tabla 1.2 Coste de los elementos hardware utilizados para el proyecto....15
Tabla 1.3 Presupuesto total del proyecto....16
Tabla 2.1 Distribución de datos que contienen la etiqueta de fiabilidad....46
Tabla 3.1 Distribución de etiquetas de fiabilidad y 5W1H del conjunto de entrenamiento....53
Tabla 3.2 Distribución de etiquetas de fiabilidad y 5W1H del conjunto de prueba....54
Tabla 4.1 Resultados obtenidos con la técnica “zero-shot”....92
Tabla 4.2 Resultados obtenidos con la técnica “few-shot”....93
Tabla 4.3 Resultados obtenidos con la técnica “CoT-zero-shot”....95
Tabla 4.4 Resultados obtenidos con la técnica “CoT-few-shot”....96
Tabla 4.5 Comparativa de resultados obtenidos con la técnica “zero-shot”....98
Tabla 4.6 Comparativa de resultados obtenidos con la técnica “few-shot”....100
Tabla 4.7 Comparativa de resultados obtenidos con la técnica “CoT-few-shot”....101
Tabla 4.8 Comparativa de resultados obtenidos con la técnica “CoT-few-shot”....103
Tabla 4.9 Resultados mejores de las técnicas....105

1. INTRODUCCIÓN

Vivimos inmersos en una transformación constante impulsada por el imparable avance de la era digital. Una era que ha transformado nuestra manera de comunicarnos y de acceder a la información. La expansión de Internet y el avance de las tecnologías móviles han sido claves en la forma en que los seres humanos nos comunicamos y recibimos información.

Actualmente la población tiene muy fácil el acceso a multitud de publicaciones digitales y a la vez se encuentra invadida por la inmensa cantidad de información que nos llega a través de distintos formatos. Como dice Alfons Cornella, “Infoxicación” [1], para aludir a la sobresaturación de información como acrónimo de intoxicación por información. La infoxicación [2] está causada por una serie de factores, entre los que se incluyen: demasiada información digital consumida a la vez en redes sociales, sitios de noticias, correos electrónicos y WhatsApp, entre otros. Esta sobrecarga agobia, cansa y desemboca en problemas que pueden estar afectando a la salud mental.

Ante este panorama nos encontramos rodeados de multitud de publicaciones envueltas en un sinfín de dudosa credibilidad. La veracidad de la información es hoy día un tema muy relevante. La preocupación por la autenticidad de los datos es algo que afecta a todos los ámbitos de nuestras vidas. Las barreras convencionales en la difusión de información y contenidos han sido prácticamente eliminadas. Este crecimiento exponencial del acceso a la información también ha dado lugar a un fenómeno preocupante: la propagación de desinformación y noticias falsas. Por ello, es importante controlar el grado de fiabilidad de la información que recibimos.

Un simple titular falso en los medios de comunicación puede dar lugar a equivocaciones y conocimientos erróneos, como por ejemplo el titular: “El presidente del gobierno, anuncia que el próximo viernes 6 de septiembre se declarará un estado de alarma de nivel 5”. La sensibilidad y gravedad de este fenómeno puede influir significativamente y llevar a tomar decisiones basadas en información no verificada.

Frente a este tipo de informaciones, tenemos la responsabilidad de actuar con filtros críticos y buscar fuentes de información fiables, especialmente en un entorno

donde cada vez es más complicado diferenciar entre la verdad de la falsedad. La principal ventaja de recurrir a fuentes de información fiables es que los datos se mantienen actualizados de forma continua, minimizando el riesgo de utilizar información obsoleta. En definitiva, debemos descartar la idea de que el fácil acceso a la información que tenemos actualmente garantiza su precisión y seguridad.

Para solucionar este problema, podemos recurrir a los grandes modelos de lenguaje, que son sistemas de inteligencia artificial (IA) diseñados para procesar y generar texto de manera que imitan la escritura humana[3]. Estos modelos, entrenados con enormes cantidades de datos, pueden analizar el contenido de las noticias y ayudar a evaluar su veracidad al identificar patrones y contextos que podrían indicar su fiabilidad o falta de ella.

En el contexto de la fiabilidad de las noticias, los modelos de lenguaje son particularmente útiles para detectar información fraudulenta. Pueden ser utilizados para evaluar la credibilidad de las fuentes, identificar manipulaciones de la información y proporcionar alertas sobre posibles noticias falsas basadas en patrones previamente identificados. Esto permite a los usuarios obtener una visión más clara y confiable de la información que consumen.

Para mejorar el rendimiento de los grandes modelos de lenguaje, se utiliza la ingeniería de prompt. La ingeniería de prompt es el proceso de diseñar y ajustar las instrucciones o preguntas que se proporcionan a un modelo de lenguaje, con el objetivo de obtener respuestas más precisas y útiles.

En este Trabajo de Fin de Grado (TFG) explicaremos lo esencial de la ingeniería del prompt y sus conceptos clave. También aplicaremos las tres técnicas más utilizadas para desarrollar prompts de forma precisa para la detección de fiabilidad en las noticias con distintos modelos de IA y veremos cómo responden.

1.1. Motivación

Hoy en día el auge digital impulsado por la IA, el desarrollo, el uso de nuevas herramientas al alcance de todo el mundo y el poder de crear cualquier noticia que

se viralice sin ser verificada por fuentes fiables me inquieta y me lleva a investigar sobre este tema.

Las noticias falsas suelen presentarse como información creíble y se propagan rápidamente debido a su capacidad para captar la atención y generar respuestas emocionales. La velocidad y viralidad en que se transmiten estas noticias en redes sociales facilitan su difusión antes de que se puedan verificar. En las plataformas digitales se expone a la sociedad a información sesgada desafiando la veracidad de la información.

Por ello este trabajo se centra en la tarea de detección de confiabilidad de las noticias y para ello se emplea un enfoque basado en la investigación de la ingeniería de prompt.

La ingeniería del prompt es una disciplina crucial para optimizar modelos de IA y la generación de lenguaje. Utilizaremos Gemini, la nueva IA generativa desarrollada por Google, como modelo generativo principal para analizar su comportamiento clasificando la veracidad de las noticias. Además, se emplea ChatGPT, la herramienta generativa más conocida, para complementar este estudio.

Actualmente, la ingeniería de prompt se encuentra en plena expansión, generando una alta demanda de profesionales especializados y es una gran oportunidad para realizar esta investigación. En este proyecto, no solo podremos integrarnos en las corrientes más avanzadas de las innovaciones tecnológicas, sino que también tendremos la oportunidad de contribuir de manera significativa a proyectos que en la actualidad están dando forma al futuro de la comunicación y la automatización.

1.2. Propósito

La idea de este proyecto es utilizar la ingeniería de prompt para crear un sistema de clasificación automática que permita determinar la fiabilidad de noticias en español para medir la credibilidad lingüística haciendo uso de modelos generativos.

Se hará uso de la técnica periodística 5W1H[4] que ayudará a identificar el contenido relevante de los titulares de las noticias. Este método es un planteamiento de preguntas clave como qué, quién, dónde, cuándo, por qué y cómo. Toda noticia necesita responder a estas preguntas para resultar completa, aunque no necesariamente fiable.

Para desarrollar este proyecto, se participará en la competición FLARES: Fine-Grained Language-Based Reliability Detection in Spanish News[5] del workshop IberLEF 2024[6], que tiene como objetivo la detección precisa de la fiabilidad de noticias escritas en español basándose en el lenguaje usado en ellas.

1.3. Objetivos

El presente TFG busca investigar los fundamentos teóricos y prácticos del aprendizaje automático para Procesamiento del Lenguaje Natural (PLN) en modelos generativos. Asimismo, pretende comprobar su potencial para mejorar la detección de fiabilidad en noticias en español utilizando algoritmos de aprendizaje profundo para la detección de patrones y la toma de decisiones basadas en datos. Con esto, los objetivos concretos son los siguientes:

- Estudiar técnicas de PLN para procesar la información de un texto.
- Estudiar las distintas aproximaciones que hay actualmente para la detección de fiabilidad en textos usando PLN.
- Analizar el conjunto de datos proporcionado en la competición FLARES del workshop IberLEF 2024.
- Realizar un estudio sobre las tres técnicas de ingeniería de prompt más utilizadas, explorando su uso y aplicación en este contexto.
- Seleccionar un entorno de desarrollo adecuado y desarrollar un sistema de clasificación automática para determinar la fiabilidad de noticias escritas en español, utilizando modelos generativos como Gemini y ChatGPT, basado en la descripción de la tarea de “Clasificación de fiabilidad” en *Kaggle*.
- Evaluar los resultados y verificar la precisión del sistema de clasificación mediante un análisis de errores y el uso de una matriz de confusión.
- Redactar una memoria del proyecto que recoja todo el trabajo realizado.

1.4. Resultados esperados

Este proyecto tiene como objetivo alcanzar una serie de resultados que no solo demuestran el éxito en la implementación técnica, sino que también aportan un análisis profundo y comparativo de las técnicas utilizadas. Entre estos resultados, se espera:

- Desarrollar con éxito la tarea FLARES: Subtarea 2 (Clasificación de confiabilidad) de la campaña IberLEF [6].
- Creación de un prompt que detecte la fiabilidad de noticias en español.
- Realización de un análisis lingüístico utilizando el sistema clasificador implementado para las noticias mediante el uso de la técnica 5W1H.
- Aprender a mejorar resultados de prompt para Gemini y ChatGPT.
- Realización de comparación exhaustiva entre las tres técnicas más usadas de prompt.
- Redacción de la memoria del proyecto donde se refleje todo el trabajo desarrollado.

1.5. Planificación

La dedicación a este TFG se ajustará a la normativa correspondiente. Teniendo en cuenta que este proyecto consta de 12 créditos ECTS supondrán un total de 300 horas de trabajo realizado.

Este proyecto se realizará durante el segundo cuatrimestre del año 2024. A continuación, en la Tabla 1.1 se muestran las actividades realizadas. Además, en la Ilustración 1.1 se presenta un diagrama de Gantt para visualizar las tareas planificadas en función del tiempo.

Actividad	**Duración**
Introducción a Python	1 semana
Aprendizaje del lenguaje de programación Python	2 semanas
Análisis del conjunto de datos para su implementación	1 semana
Estudio sobre consejos generales para diseñar prompts efectivos	1 semana
Estudio sobre la aplicación de las técnicas de ingeniería de prompt al proyecto	1 semana
Estudio de las métricas de evaluación e implementación del sistema	2 semanas
Ejecución completa del conjunto de datos de prueba de las tres técnicas de prompt	6 semanas
Ejecución completa del conjunto de datos de test para su evaluación de cada una de las tres técnicas	2 semanas
Análisis y valoración del desempeño de los sistemas implementados	1 semana
Conclusión general del proyecto	1 semana
Redacción de la documentación completa del trabajo	18 semanas

Tabla 1.1 Planificación del proyecto

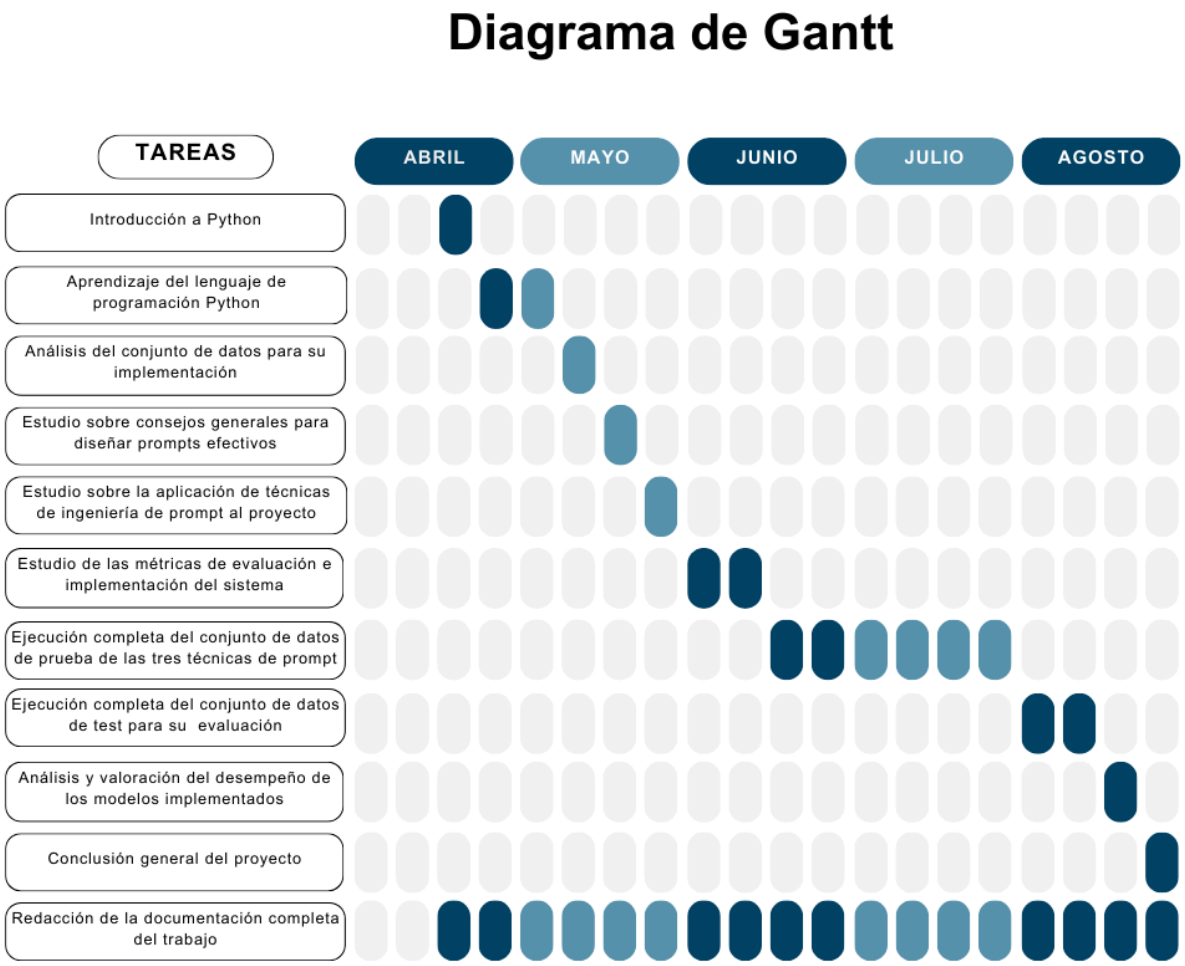

Ilustración 1.1 Diagrama de Gantt de la planificación del proyecto

1.6. Presupuesto

En este apartado se realizará una estimación de costes del proyecto en el que se mostrarán los precios de las licencias y aplicaciones utilizadas, así como la utilización de servicios, costes hardware, costes en recursos humanos y costes no previstos para la realización del proyecto.

Para el desarrollo del código, se ha utilizado *Jupyter Notebook*, una potente herramienta de código abierto. Esta aplicación web permite a los desarrolladores crear y compartir documentos interactivos que integran código, texto, fórmulas y contenido multimedia, lo que la convierte en una herramienta invaluable para la programación, el análisis de datos y la ciencia de datos.

También, se ha empleado Google Colab, una plataforma basada en *Jupyter Notebook* que no requiere configuración previa y ofrece acceso gratuito a recursos

de computación en la nube. Es una herramienta adecuada para el aprendizaje automático y la ciencia de datos que me podrá ser de utilidad en un futuro próximo.

Para la evaluación, se ha utilizado Visual Studio Code, un entorno de desarrollo integrado (IDE) esencial por sus múltiples funcionalidades y extensiones. Gracias a la licencia académica gratuita de la Universidad de Jaén, he tenido acceso completo a todas sus características avanzadas. Este es un editor de código fuente desarrollado por Microsoft para Windows, Linux, MacOS y Web que incluye soporte para la depuración de código, control integrado de Git, resaltado de sintaxis y refactorización de código entre otros.

El proyecto se empezó a implementar en Google Colab, pero finalmente se realizó el sistema de detección de noticias con Jupyter Notebook. Opté por Jupyter Notebook porque se ejecuta localmente y, después de comparar los recursos de las máquinas, descubrí que mi equipo local ofrecía un rendimiento superior al de Google Colab, lo que resultó en una ejecución más rápida y eficiente del sistema.

La tarifa de internet es fundamental para conectar el ordenador a la red y realizar la investigación de manera eficiente. Este servicio mensual tuvo un costo fijo de **33,95 €.**

El consumo de electricidad experimenta cambios dependiendo de factores económicos y la oferta y demanda de la energía eléctrica. El precio medio total de la electricidad para los meses de abril, mayo, junio, julio y agosto de 2024 es aproximadamente **55,01 €/MWh.**

No se debe olvidar el coste del equipo de trabajo a contratar. Para realizar el proyecto contaremos con un ingeniero de prompts. La jornada laboral estándar es de 40 horas semanales, lo que equivale a unas 160 horas mensuales y un salario medio de 2,000 euros brutos mensuales. Como he dedicado un tiempo de 300 horas al proyecto, el coste del personal será de (**2000€ * 5 meses = 10.000€**).

Una vez analizados los costes de software, costes de consumo de electricidad y los costes de la mano de obra, se procederá a calcular el coste del hardware. Está formado por los dispositivos utilizados para la realización del proyecto. Para ello, se ha elegido un portátil *AMD Ryzen 7 5800H with Radeon Graphics* con un precio de **949,00€**. También, cuento con un monitor *Lenovo C24-25*

valorado en **169,00€** y un ratón *Razer Deathadder Essentia*l de **24,90€**. Ahora, se presenta una tabla resumen que detalla los componentes hardware, incluyendo su precio inicial, vida útil estimada, amortización mensual y el coste final.

Elemento	Vida útil estimada	Precio inicial	Amortización mensual	Coste final
AMD Ryzen 7 5800H with Radeon Graphics	6 años	949,00€	13,18€	**65,90€**
Monitor Lenovo C24-25	5 años	169,00€	2,82€	**14,10€**
Razer Deathadder Essential	4 años	24,90€	0,52€	**2,60€**
Suma total				**82,60€**

Tabla 1.2 Coste de los elementos hardware utilizados para el proyecto

Después de calcular el coste de cada uno de los servicios utilizados, procedemos a calcular el coste total. Sumando todos los costes software, hardware y de servicios obtenemos un presupuesto final de **10.527,40€** para realizar el proyecto. En la Tabla 1.3 se muestra el desglose.

Servicios	Coste total
Visual Studio Code	**0€**
Google Colab	**0€**
Jupyter Notebook	**0€**
Servicio de internet	**33,95€ * 5 meses = 169,75€**
Consumo de luz	**55,01€/mes * 5 meses = 275,05€**
Ingeniero de prompt	**2.000€/mes * 5 meses = 10.000€**
Equipo informático	**82,60€**
Suma total	**10.527,40€**

Tabla 1.3 Presupuesto total del proyecto

1.7. Estructura del proyecto

A continuación, se detalla un esquema general del proyecto donde se describen los contenidos de cada una de las principales secciones.

Introducción

En este primer capítulo, se introduce el problema que se abordará durante el proyecto, así como los objetivos y resultados esperados una vez terminado el TFG. Asimismo, se incluye una estimación del presupuesto necesario para realizarlo.

Antecedentes

Este capítulo proporciona un marco teórico y contextual para el desarrollo del sistema de clasificación para la detección de fiabilidad en noticias en español. Se mencionan conceptos clave del PLN así como las técnicas para el preprocesamiento del texto. Luego, se definen las técnicas para el preprocesamiento de un texto y se introduce la Ingeniería de Prompt, definiendo su utilidad, su función y desglosando sus elementos principales.

También se presentan las tres técnicas más usadas y su potencial en este ámbito, además de mostrar su utilidad en el proyecto. Asimismo, se abordan los tipos, aplicaciones e importancia de la ingeniería de prompt.

Además, se discute la importancia de la fiabilidad en el lenguaje, especialmente en el contexto de las "fake news" y la guía RUN-AS[7] que sirve como referencia para la evaluación del sistema. Por último, se describe la campaña de evaluación IberLEF y la tarea FLARES en la que se ha participado. También se describe el corpus o conjunto de datos proporcionado y las medidas de evaluación que utilizaremos.

Desarrollo

En este capítulo se describe todo el proceso del desarrollo del sistema de clasificación para la detección de fiabilidad en noticias en español. Se abordarán las principales etapas del proyecto, desde el análisis del conjunto de datos hasta la implementación de los sistemas y las técnicas de evaluación.

Se explicarán detalladamente cada una de las técnicas de prompting utilizadas. También se discuten los hiperparámetros configurados y se explican los dos modelos generativos aplicados: Gemini y ChatGPT. Finalmente, se describe el post-procesamiento tras la ejecución de los modelos y las tecnologías y entornos de desarrollo que han facilitado la implementación del sistema.

Resultados y discusión

En este cuarto capítulo se analizan los resultados obtenidos durante el desarrollo del sistema. Se incluye un análisis detallado de los errores en los que el

sistema no ha clasificado correctamente la fiabilidad de las noticias y, una comparación de resultados obtenidos por los demás participantes en la tarea FLARES (Clasificación de confiabilidad). Además, se analizarán los errores de ejecución que hemos encontrado al desarrollar el sistema.

Conclusiones

En este capítulo se presentan las conclusiones sobre los resultados finales obtenidos reflexionando sobre el desempeño del sistema. Se proponen posibles mejoras al sistema de detección de fiabilidad en las noticias, trabajos futuros para ampliar y perfeccionar el proyecto y la satisfacción personal.

Anexos

Este sexto capítulo presenta dos anexos: en el primero se incluyen tablas comparativas de prompt de todos los modelos del sistema desarrollados y, en el segundo, se presentan los manuales de instalación y de usuario

Bibliografía

En este último capítulo, se nombran todas las referencias y citas bibliográficas que han permitido ampliar la información y los conocimientos que se plantean durante todo el proyecto.

2. ANTECEDENTES

Este proyecto propone clasificar noticias según su fiabilidad empleando diversas técnicas de ingeniería de prompts. Para ello, utilizaremos modelos generativos avanzados para analizar y categorizar el contenido de las noticias, aprovechando su capacidad para entender y generar lenguaje humano de manera efectiva. Para lograr esto, es fundamental comprender el concepto de PLN y cómo las máquinas interpretan y producen texto.

La ingeniería de prompts será clave para afinar los modelos generativos, mejorando su precisión y relevancia en la clasificación de noticias.

2.1. Procesamiento del Lenguaje Natural (PLN)

La IA es en la actualidad una disciplina con numerosas aplicaciones en el ámbito educativo. Desde un punto de vista técnico, se define como herramientas que optimizan procesos y mejoran la resolución de problemas, gracias a su habilidad para manejar grandes tareas.

En el avance de este campo surge el PLN, una rama del conocimiento que se encuentra en la intersección de la IA y la lingüística. El objetivo principal es dotar a los sistemas automáticos de la capacidad de extraer y clasificar información del lenguaje humano. Se centra en el análisis de las comunicaciones humanas y, en concreto, de su lenguaje, llegando a transformar el lenguaje natural en lenguaje formal (lenguaje de programación) que los ordenadores pueden procesar.

Los sistemas de PLN analizan grandes cantidades de datos lingüísticos para identificar patrones y comportamientos a través del aprendizaje automático. Esto permite que las máquinas no sólo comprendan el lenguaje, sino que también generen respuestas coherentes. El PLN[8] es el proceso por el cual las máquinas convierten los datos en texto permitiendo que las computadoras comuniquen ideas con frases que parecen haber sido escritas por humanos. Esto implica dos procesos fundamentales:

- **Comprensión del lenguaje natural:** Mediante modelos de PLN, las máquinas decodifican las consultas humanas, identificando intenciones y extrayendo información relevante.
- **Respuesta en lenguaje natural:** Una vez procesada la solicitud, la máquina genera respuestas utilizando NLG que sean naturales y fáciles de comprender.

Diversas tecnologías utilizan PLN, entre las cuales se encuentran los modelos generativos de IA que aplicaremos para alcanzar los objetivos de este proyecto. Los modelos generativos son algoritmos de aprendizaje automático diseñados para generar texto, imágenes u otros datos basados en patrones aprendidos a partir de grandes cantidades de información. Estos modelos pueden producir contenido

nuevo y coherente al interactuar con datos de entrada, lo que los hace valiosos para tareas como la generación de texto y la clasificación de noticias.

En el ámbito del PLN, existen varios tipos de modelos generativos, incluidos los modelos de lenguaje autorregresivos, los basados en transformers, los basados en autoencoders, los modelos generativos adversariales (GANs), los autoencoders variacionales (VAEs), los modelos de difusión, los modelos generativos para audio y los modelos multimodales.

Para este proyecto, se ha seleccionado Gemini y ChatGPT, ambos modelos de lenguaje autorregresivos basados en transformers. Los modelos autorregresivos generan texto prediciendo una palabra a la vez, utilizando las palabras anteriores en la secuencia como contexto para cada predicción subsiguiente. Esta técnica permite crear texto fluido y coherente al construir una secuencia de palabras de manera incremental.

Seleccionamos la herramienta Gemini porque es la herramienta más moderna en el momento de iniciar este TFG. Su tecnología de vanguardia ofrece capacidades avanzadas en generar texto predictivo con alta precisión. Por otro lado, ChatGPT se elige por su amplia utilización y popularidad hoy en día, destacando por su robustez y eficacia en mantener conversaciones naturales y contextualmente apropiadas. Esta combinación permitirá una clasificación más precisa y eficiente de noticias según su fiabilidad, aprovechando lo mejor de ambos modelos generativos.

El PLN y los prompts están interconectados de manera fundamental. Los prompts sirven como una interfaz crítica que permite a los modelos de PLN generar respuestas relevantes y contextualmente adecuadas, optimizando la interacción humano-máquina y ampliando las aplicaciones prácticas de la IA en el manejo del lenguaje natural.

2.2. Técnicas para el preprocesamiento de un texto

El preprocesamiento de texto en el ámbito de la detección de fiabilidad de noticias es crucial para asegurar que los datos sean consistentes y útiles para el análisis. A continuación, se describen algunas técnicas clave de preprocesamiento:

- **Tokenización**: este proceso implica dividir el texto en unidades más pequeñas llamadas tokens. En este contexto, la tokenización divide el texto de las noticias en unidades manejables, que pueden ser palabras, frases o caracteres facilitando la identificación de patrones y términos.

- **Lematización y stemming**: ambas técnicas se utilizan para reducir los tokens a su forma base o raíz. Esto significa que palabras como "verificar", "verificación" y "verificado" se pueden tratar como una única unidad. Esto es útil para identificar términos relevantes sobre la fiabilidad de una noticia, tales como aquellos relacionados con la validación de hechos.

- **Eliminación de stopwords**: Las stopwords son palabras comunes que no aportan valor significativo al análisis, como "y", "el", "de". Eliminarlas ayuda a reducir el ruido en los datos y a enfocarse en las palabras que realmente aportan información relevante para el análisis. También aumenta el rendimiento del modelo ya que se reduce el número de tokens.

- **Normalización**: Este proceso incluye varias técnicas para estandarizar el texto. Puede incluir la conversión de toda la noticia a minúsculas para evitar que las mismas palabras en diferentes casos se traten como distintas, así como la eliminación de caracteres especiales y la corrección de errores ortográficos.

- **Codificación de texto**: En algunos casos, es necesario convertir el texto en representaciones numéricas para que pueda ser procesado por algoritmos de aprendizaje automático. Esto puede hacerse mediante técnicas como la representación en bolsa de palabras (bag-of-words) que cuentan la frecuencia de palabras, la representación TF-IDF (Term Frequency-Inverse Document Frequency) que pondera estas frecuencias para destacar términos clave o el uso de embeddings de palabras como Word2Vec o GloVe que capturan significados semánticos.

En este TFG se han empleado modelos generativos avanzados como Gemini y ChatGPT, que incorporan técnicas avanzadas de PLN para mejorar el análisis y la clasificación de textos. Estos modelos aplican la tokenización para dividir el texto en unidades manejables, la lematización y el stemming para reducir las palabras a sus

formas base, y la eliminación de stopwords para filtrar palabras comunes que no aportan valor informativo. Además, realizan la normalización del texto para estandarizarlo, lo que incluye convertir caracteres a minúsculas y eliminar acentos. También se utiliza la codificación de texto para representar palabras y frases en formatos numéricos que los modelos pueden procesar eficientemente.

2.3. Ingeniería de prompt

Para contextualizar este proyecto, comenzaremos con un estudio sobre la ingeniería de prompt, explorando su historia, clasificación y aplicaciones.

Actualmente, la IA está revolucionando diversos aspectos de nuestra vida cotidiana, desde la interacción con la tecnología hasta la toma de decisiones basadas en datos. En el centro de esta transformación se encuentra una técnica esencial para la optimización y efectividad de los modelos de IA: la ingeniería de prompt.

La ingeniería de prompt consiste en diseñar y perfeccionar las instrucciones que se proporcionan a un modelo de IA para que genere resultados precisos y útiles. En un mundo donde los modelos de lenguaje como GPT-4 de OpenAI y Gemini de Google juegan roles cada vez más decisivos, la habilidad de guiar a estos sistemas con instrucciones claras y efectivas resulta indispensable.

Esta disciplina no solo mejora la precisión y relevancia de las respuestas generadas por la IA, sino que también permite a usuarios y desarrolladores personalizar las aplicaciones de IA según sus necesidades específicas.

Además, la ingeniería de prompt es fundamental para resolver problemas complejos. Al diseñar prompts que tengan en cuenta diversos contextos culturales y lingüísticos, los modelos de IA pueden ajustarse de manera más efectiva para atender a cualquier tarea que el usuario pueda necesitar ofreciendo soluciones precisas y personalizadas.

Su importancia radica en su capacidad para convertir la IA en una tecnología verdaderamente útil y accesible, capaz de resolver problemas reales y mejorar nuestra interacción con el mundo digital.

2.3.1. Definición, utilidad y función del prompt

Un prompt es una instrucción o conjunto de instrucciones proporcionadas a un modelo de lenguaje para generar una respuesta específica. Es el mensaje a través del cual el usuario se comunica con el ordenador solicitando una tarea concreta. En esencia, es un punto de entrada de una petición que permite al usuario dirigir las operaciones realizadas por el sistema.

El prompt es la herramienta a través de la cual podemos comunicarnos de manera eficaz con la máquina. Dicho de otro modo, es la instrucción que usamos para consultar, pedir o guiar a la IA según lo que necesitemos.

La función principal de prompt es dirigir y controlar la generación de respuestas por parte del modelo de lenguaje. Mediante la instrucción original, se especifica el tema y el enfoque requerido. Adicionalmente, es posible definir el tono y el estilo de la respuesta, permitiendo su adaptación a diferentes audiencias.

Es fundamental delimitar claramente la solicitud para minimizar ambigüedades y mejorar la precisión de los resultados. Los prompts bien formulados también permiten una mayor personalización de las interacciones del usuario, asegurando que las respuestas sean más pertinentes y específicas.

2.3.2. Elementos principales

A la hora de crear prompts en lenguaje natural para los chatbots de IA, es crucial comprender los elementos fundamentales para lograr el resultado deseado. Los elementos de prompt guían al modelo de IA en la dirección deseada, garantizando claridad y contexto. Estos son algunos de los principales elementos que conforman un prompt.

- **Instrucción:** Una tarea o acción específica que se desea que el modelo realice. Las instrucciones pueden variar en complejidad desde simples tareas hasta tareas más elaboradas. Las instrucciones claras y detalladas ayudan al modelo a entender exactamente lo que se espera de él.

- **Contexto**: Preparar el contexto idóneo es importante para orientar las respuestas del modelo generativo de manera precisa y relevante. Esto significa que la IA necesita pistas sobre el tema específico de nuestra consulta para generar una respuesta adecuada.

- **Datos de entrada**: Son la información inicial que se proporciona al modelo de IA para que pueda procesarla y ofrecer una salida apropiada. Estas entradas pueden ser preguntas directas, fragmentos de texto o contexto adicional.

- **Indicador de salida:** Especifica el tipo o formato de la respuesta esperada. Es fundamental para dirigir cómo el modelo debe organizar y presentar la información solicitada. Establecer claramente el indicador de salida ayuda al sistema estructurar su respuesta de manera coherente y oportuna, especialmente en tareas que necesitan cumplir con formatos específicos o estructuras particulares para satisfacer las necesidades del usuario.

No todos los componentes son necesarios para solicitar un prompt y el formato depende de la tarea en cuestión. Como hemos mencionado anteriormente, un prompt bien estructurado puede mejorar significativamente la calidad de las soluciones generadas por la IA y pueden ayudar a minimizar la ambigüedad y mejorar la precisión de las respuestas.

Para mejorar la interacción contextual del modelo, es esencial seguir estas otras pautas:

- Plantear el tono adecuado de la respuesta nos posibilita indicar la emoción que queremos recibir. Por ejemplo, si pedimos a la IA que haga una crítica profesional de un diseño de página web, queremos que la respuesta tenga un tono serio y experto, como si alguien con mucha experiencia diera su opinión.

- En cuanto a la longitud de la respuesta es importante especificar claramente la profundidad y detalle de la solución, adaptada a los requisitos específicos que el usuario necesita.

2.3.3. Técnicas

Optimizar los prompts contribuye a obtener resultados mejores en diversas tareas, lo cual es el concepto central de la ingeniería de prompts. En este apartado, abordaremos las tres técnicas avanzadas de ingeniería de prompts más utilizadas que nos permiten realizar tareas más complejas e interesantes.

2.3.3.1. Prompt sin entrenamiento previo (Zero-shot)

Los modelos de lenguaje actuales han sido entrenados con grandes volúmenes de datos y ajustados para seguir instrucciones específicas. Gracias a este extenso entrenamiento, pueden realizar una variedad de tareas sin necesidad de entrenamiento adicional.

El ajuste de instrucciones ha demostrado ser muy efectivo para mejorar el aprendizaje sin la necesidad de entrenamiento adicional, tal como se describe en el estudio de Jason Wei titulado *"Fine Tuned Language Models are Zero-Shot Learners"*[9]. Este enfoque implica la adaptación de modelos de lenguaje mediante el uso de conjuntos de datos detallados que incluyen instrucciones específicas. Al proporcionar directrices concretas y precisas, los modelos pueden ejecutar tareas sin requerir un entrenamiento previo para cada tarea específica, lo que se conoce como capacidad "zero-shot".

Una parte clave de este proceso es la implementación del aprendizaje por refuerzo a partir de la retroalimentación humana (RLHF)[10]. Este método permite que los modelos ajusten sus respuestas basándose en la retroalimentación recibida de los usuarios, lo que ayuda a alinear mejor los resultados con las preferencias humanas. A través de RLHF, los modelos no solo mejoran en términos de precisión y relevancia, sino que también se vuelven más intuitivos y satisfactorios para los usuarios.

Cuando el prompt sin entrenamiento no da buenos resultados, se sugiere proporcionar demostraciones o ejemplos dentro de la instrucción, lo que se conoce como entrenamiento con pocos ejemplos.

Un ejemplo de prompt utilizando esta técnica podría ser el siguiente:

```
Python
Prompt: "Evalúa la fiabilidad del siguiente texto. Clasifica como
'confiable', 'semiconfiable', o 'no confiable' basándote en la precisión y
objetividad del mismo.

Texto a evaluar: ‘La acrilamida, una vez en el organismo, se transforma en
glicidamida.’"
```

2.3.3.2. Prompt con pocas muestras (Few-shot)

Aunque los grandes modelos de lenguaje demuestran capacidades impresionantes sin entrenamiento previo (zero-shot), tienen limitaciones al abordar tareas más complejas. Para superarlas, los prompts con pocas muestras se han convertido en una técnica útil para facilitar el aprendizaje contextual. Al incorporar demostraciones dentro del prompt, podemos orientar al modelo para que mejore su rendimiento. Estas demostraciones funcionan como ejemplos que condicionan al modelo a generar mejores respuestas en situaciones posteriores.

Según Touvron et al. (2023) en su estudio *"LLaMA: Open and Efficient Foundation Language Models"* [11], esta técnica de prompt surgió cuando los modelos alcanzaron un tamaño considerable, como se describe en el trabajo de Kaplan et al. (2020) *"Scaling Laws for Neural Language Mode*ls" [12].

En el siguiente artículo de Min et al. (2022) en *"Rethinking the Role of Demonstrations: What Makes In-Context Learning Wor*k?" [13], se presentan algunos consejos sobre el uso de demostraciones con pocas muestras:

- **Espacio de etiquetas y distribución del texto de entrada**: Es importante especificar claramente las etiquetas y la distribución del texto de entrada en los ejemplos, incluso si las etiquetas no son siempre correctas para las

entradas individuales. La presencia de etiquetas, aunque sean incorrectas, ayuda a guiar al modelo.

- **Formato de las demostraciones**: El formato utilizado en las demostraciones influye significativamente en el rendimiento del modelo. Incluso las etiquetas seleccionadas al azar pueden resultar más beneficiosas que la ausencia total de ellas.
- **Selección de etiquetas**: Al escoger etiquetas de una distribución real en lugar de una uniforme, se puede mejorar el desempeño del modelo.

Estas sugerencias no solo mejoran la capacidad de los modelos para aprender del contexto presentado, sino que también los hacen más robustos y exactos en sus respuestas.

Los prompts que utilizan un número reducido de ejemplos son eficaces para muchas tareas, aunque no son infalibles, ya que tienen limitaciones especialmente en tareas de razonamiento más complejas.

Una estrategia útil podría ser descomponer el problema en pasos y demostrar cada parte al modelo, llamada prompt de cadena de pensamientos, que ha ganado mucha popularidad gracias a sus buenos resultados.

Un ejemplo de prompt sobre cómo aplicar esta sería el siguiente:

```python
Prompt: "Evalúa la fiabilidad del siguiente texto. Clasifica como 'confiable', 'semiconfiable', o 'no confiable' basándote en la precisión y objetividad del mismo.

Ejemplo 1: Texto a evaluar: 'La Tierra es plana y está sostenida por elefantes.' Respuesta esperada: 'no confiable'

Ejemplo 2: Texto a evaluar: 'El agua hierve a 100 grados Celsius a nivel del mar.' Respuesta esperada: 'confiable'

Texto a evaluar: ‘La acrilamida, una vez en el organismo, se transforma en glicidamida.'"
```

2.3.3.3. Prompt cadena de pensamiento (CoT)

El método de prompting, conocido como Chain of Thought (CoT) es una técnica innovadora que estimula a los modelos de lenguaje a explicar su razonamiento. La técnica se centra en ofrecer ejemplos que ilustran el proceso de razonamiento, lo que motiva al sistema a explicar su propio razonamiento al responder a una solicitud. Esta explicación detallada suele llevar a resultados más precisos.

Se ha demostrado que este tipo de prompt es efectivo para mejorar los resultados en tareas de aritmética, sentido común y razonamiento simbólico. Según Jason Wei, es importante destacar como limitación que "CoT solo produce mejoras de rendimiento con modelos de alrededor de 100 mil millones de parámetros". Los modelos más pequeños generaron cadenas de pensamiento ilógicas, lo que resultó en una precisión inferior a la de prompting estándar. Las mejoras en el rendimiento con CoT suelen ser proporcionales al tamaño del modelo.

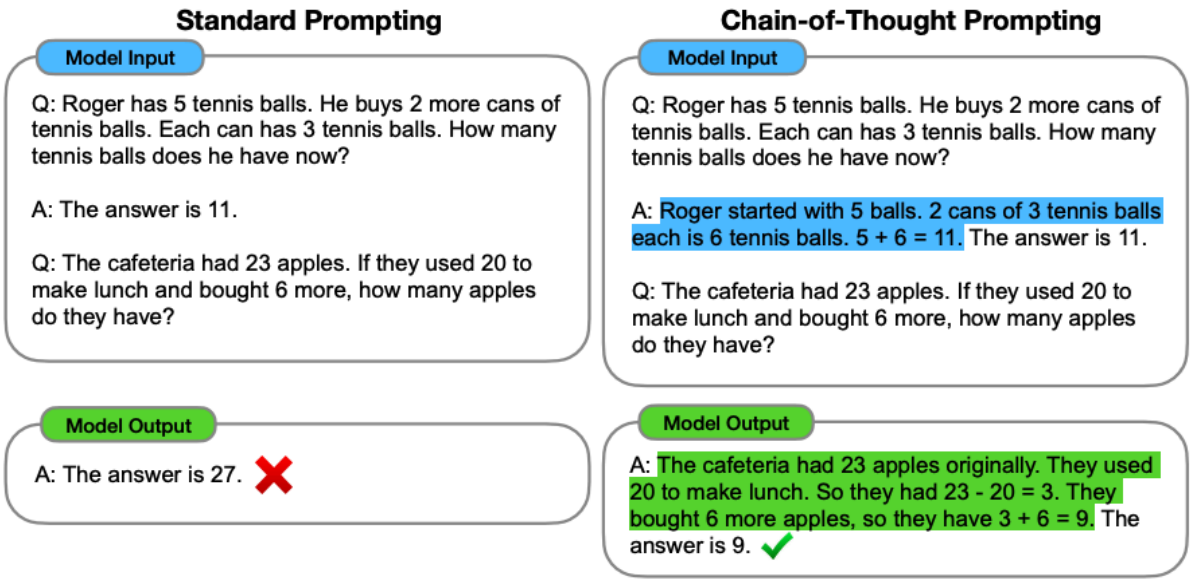

Ilustración 2.1 Ejemplo de mejora de rendimiento utilizando cadena de pensamiento.

Fuente: Guía de Prompting [14]

El prompt por cadena de pensamientos (CoT) permite capacidades de razonamiento complejas a través de pasos de razonamiento intermedios. Se puede

combinar con prompts de pocas muestras (few-shot) para obtener mejores resultados en tareas más complejas que requieren razonamiento antes de responder.

Más adelante, evaluaremos cuál de las tres técnicas para nuestro modelo de IA, que clasifica el texto en tres etiquetas, resulta ser la más eficaz.

Un ejemplo de prompt aplicando esta técnica sería el siguiente:

```
Python
Prompt: "Evalúa la fiabilidad del siguiente texto. Clasifícalo como
'confiable', 'semiconfiable' o 'no confiable' basándote en la precisión y
objetividad del mismo. Piensa paso a paso para llegar a la respuesta final.

Ejemplo 1: Texto a evaluar: 'La Tierra es plana y está sostenida por
elefantes.' Paso 1: Considerar la precisión: La afirmación es incorrecta
según el conocimiento científico actual. Paso 2: Evaluar la objetividad: No
se presenta evidencia objetiva para apoyar la afirmación. Paso 3: Conclusión:
Esta información no es confiable. Respuesta esperada: 'no confiable'

Ejemplo 2: Texto a evaluar: 'El agua hierve a 100 grados Celsius a nivel del
mar.' Paso 1: Considerar la precisión: La afirmación es correcta y está
científicamente comprobada. Paso 2: Evaluar la objetividad: La información es
presentada de manera neutral, sin sesgos ni opiniones personales. Paso 3:
Conclusión: Esta información es confiable. Respuesta esperada: 'confiable'

Texto a evaluar: 'La acrilamida, una vez en el organismo, se transforma en
glicidamida.'"
```

2.3.4. Aplicaciones y tipos de prompt

La ingeniería de prompts abarca una variedad de aplicaciones y se centra en la creación y el ajuste estratégico de instrucciones específicas para mejorar la interacción con los modelos de IA. Por ello, cada tipo de prompt está relacionado con la aplicación a la que vaya destinada. Entre sus aplicaciones más destacadas se encuentran:

- **Resumen de textos:** es una aplicación fundamental y versátil en el ámbito de la IA. Los modelos de lenguaje pueden sintetizar grandes volúmenes de

información en resúmenes concisos y claros, lo que es particularmente útil en sectores como el periodismo, la educación y la investigación. Por ejemplo, pueden convertir un artículo científico extenso en un breve resumen que destaque los hallazgos clave, o reducir un informe financiero detallado a unos pocos párrafos que resaltan las tendencias más importantes. Esta capacidad de resumir eficientemente no solo ahorra tiempo a los lectores, sino que también facilita la comprensión y el análisis de la información esencial. El tipo de prompt que usaremos para esta finalidad será el **prompt de resumen,** que consiste en proporcionar un texto completo y pedirle a la IA que genere un resumen del mismo.

- **Extracción de información:** Los modelos de lenguaje son herramientas poderosas para la extracción de información relevante de grandes conjuntos de datos. Además de generar lenguaje natural, son altamente competentes en tareas de clasificación y PLN, como la identificación de entidades mencionadas en un texto, la extracción de fechas, nombres, ubicaciones, y otras categorías específicas de información. Por ejemplo, pueden analizar miles de documentos legales para encontrar cláusulas específicas, o revisar grandes cantidades de correos electrónicos para extraer datos de contacto. Esta capacidad de extraer información de manera precisa y rápida es inestimable en campos como la medicina, el derecho y el análisis de mercado.

- **Preguntas y respuestas:** Para obtener respuestas precisas y relevantes, es importante mejorar el formato de prompt. Un prompt bien estructurado puede combinar instrucciones claras, contexto adecuado e indicadores de entrada y salida específicos para guiar al modelo hacia respuestas más exactas. Por ejemplo, en lugar de simplemente preguntar "¿Cúal es la estructura más utilizada en Python?", un prompt más estructurado podría ser: "Basado en la información proporcionada en el párrafo anterior, ¿puedes indicar cúal es la estructura más utilizada en Python?". Este tipo de prompts detallados ayuda a reducir la ambigüedad y mejora la precisión de las respuestas. Aunque los componentes adicionales no son estrictamente necesarios, son prácticas recomendables para obtener resultados más específicos y útiles. El tipo de prompt usado es el **prompt de pregunta y respuesta**, el cual se basa en

proporcionar una pregunta a la IA y pedirle que genere una respuesta coherente.

- **Clasificación de texto:** La clasificación de texto va más allá de dar instrucciones simples; requiere una comprensión profunda del contexto y la utilización de diversos elementos en el prompt, como ejemplos concretos y datos de entrada. Por ejemplo, en este proyecto, en lugar de simplemente instruir al modelo a clasificar las noticias en tres etiquetas diferentes “confiable”, “semiconfiable” o “no confiable” según el conocimiento previo que posee, podemos proporcionar ejemplos de noticias etiquetadas correctamente y detallar las características que definen cada categoría. Esta es la técnica anteriormente mencionada de "few-shot learning" que mejora la precisión del modelo, especialmente en tareas complejas donde las instrucciones por sí solas no son suficientes. Para esta finalidad se usa el **prompt de clasificación,** los usamos para entrenar modelos de lenguaje natural en clasificación de textos. Consiste en proporcionar al modelo un texto y pedirle que determine a qué clase o categoría pertenece. Este tipo de prompt será el utilizado en este trabajo.

- **Conversación:** Enseñar a los modelos de lenguaje a comportarse de manera coherente y mantener una identidad específica a lo largo de una conversación es una de las aplicaciones más interesantes de la ingeniería de prompts. Esto es especialmente útil para desarrollar chatbots de atención al cliente que puedan proporcionar respuestas técnicas y científicas de manera precisa. Por ejemplo, un chatbot diseñado para asistencia técnica debe no solo entender preguntas específicas, sino también mantener un tono profesional y ofrecer soluciones detalladas. Esta técnica, conocida como "role prompting", permite crear sistemas conversacionales que simulan interacciones humanas con un alto grado de fidelidad y utilidad.

- **Generación de código:** La generación de código es una de las aplicaciones más impresionantes de los modelos de lenguaje. Herramientas como Copilot han demostrado la capacidad de estos modelos para asistir en tareas de programación, desde escribir funciones simples hasta generar bloques de código más complejos basados en descripciones en lenguaje natural. Por

ejemplo, un prompt que describe una función matemática puede resultar en un fragmento de código completo en Python o JavaScript. Esta capacidad no sólo acelera el desarrollo de tareas, sino que también libera la programación al hacerla accesible a personas con menos experiencia técnica.

- **Razonamiento**: El razonamiento sigue siendo uno de los desafíos más grandes para los modelos de lenguaje. Aunque han mostrado avances en tareas aritméticas y de lógica, los modelos actuales aún enfrentan dificultades con problemas que requieren múltiples pasos de razonamiento. Por ejemplo, resolver problemas matemáticos complejos o inferir conclusiones lógicas a partir de datos no triviales. Este tipo de tareas requiere técnicas avanzadas de ingeniería de prompts, como desglosar el problema en pasos más pequeños y proporcionar múltiples ejemplos de razonamiento intermedio. Aunque estos métodos están en desarrollo, ya podemos ver mejoras significativas en la capacidad de los modelos para manejar tareas de razonamiento básico, lo que abre la puerta a aplicaciones más avanzadas en el futuro.

Otros tipos de prompt comúnmente usados son:

- **Prompt de texto completo**: consiste en proporcionar al modelo una frase o párrafo para que lo complete o genere más texto a partir de este. Este tipo se utiliza comúnmente en la generación de texto creativo, la creación de historias o en la generación de diálogos.
- **Prompt de corrección:** Se utiliza para la corrección gramatical y ortográfica. Se le proporciona a la IA un texto que contenga errores gramaticales o de ortografía y se le solicita que lo corrija.
- **Prompt de predicción de palabras:** Se aplica comúnmente en la corrección de textos y en la predicción de texto en tiempo real.
- **Prompt de elección múltiple:** podemos formular una pregunta o afirmación acompañada de varias opciones de respuesta, y pedirle que seleccione la opción correcta.

2.3.5. Importancia del prompt

El prompt es clave para dirigir y limitar las respuestas de los modelos de lenguaje. Permite a los usuarios especificar el tipo de resultados que desean, ajustando tanto el formato como la extensión de las respuestas. Esto facilita una interacción más intuitiva y eficiente con la tecnología.

Los usuarios pueden obtener información precisa y relevante mediante el uso de prompts, mejorando la efectividad de sus consultas y optimizando el tiempo de búsqueda y procesamiento de datos. Además, el uso de prompts fomenta el pensamiento crítico y la creatividad, al requerir formulaciones claras y proporcionar acceso a un gran conocimiento.

La utilización de prompts no solo sirve como herramienta técnica, sino que también tiene una dimensión inspiradora al permitir la interacción con un lenguaje más sofisticado y creativo. Esta capacidad expande los límites del entendimiento y la expresión, enriqueciendo las experiencias de aprendizaje y comunicación humana en el ámbito de la tecnología moderna.

2.3.6. Aplicación en el proyecto

Hemos comentado las aplicaciones de la ingeniería de prompt y, a continuación, explicaremos cómo se puede aplicar a la detección de fiabilidad en las noticias en español.

Antes de la aparición de las herramientas avanzadas de IA y aprendizaje automático, la detección de noticias falsas se realizaba principalmente mediante métodos manuales y basados en el juicio humano. Se verificaba el contenido, el medio de comunicación, el autor, las fuentes, las imágenes, entre otros. Estos enfoques ayudaban a identificar y corregir información falsa, aunque eran limitados por el tiempo y los recursos disponibles.

Desde la aparición de la IA, la detección de noticias falsas ha evolucionado significativamente, incorporando herramientas avanzadas y técnicas automatizadas para abordar el problema de manera más eficiente.

Entre los diversos enfoques existentes como análisis automatizado de texto, verificación de hechos automatizada, análisis de imágenes y videos, análisis de redes sociales, evaluación de la credibilidad de las fuentes, modelos predictivos y de clasificación, se destaca el enfoque de clasificación ternaria que es el utilizado en esta investigación.

El enfoque de clasificación ternaria se utiliza para categorizar las noticias o la información en tres posibles categorías: "confiable", "semiconfiable" y "no confiable". Fue propuesto por investigadores como Rubin, quien sugirió clasificar las noticias en tres categorías: verdadera, falsa y mixta o engañosa para capturar mejor los matices de la información en la investigación "*Deception detection for news: Three types of fakes*"[15]. Esta orientación ha sido adoptada en estudios académicos y competiciones como el "Fake News Challenge"[16] y "FEVER"[17], proporcionando un método más preciso para evaluar la credibilidad de los textos.

La clasificación ternaria se basa en técnicas de aprendizaje automático supervisado, donde el modelo es entrenado utilizando un conjunto de datos previamente etiquetado con estas tres categorías. Durante el entrenamiento, el modelo aprende a identificar patrones, características lingüísticas, fuentes, y otros elementos distintivos que ayudan a determinar el nivel de confiabilidad de una noticia o declaración.

La razón por la que se emplea el enfoque de clasificación ternaria es porque proporciona una información más matizada y permite una mayor flexibilidad al abordar la detección de noticias falsas [18].

En este TFG, primero analizamos el conjunto de datos disponibles en la competición "FLARES: Clasificación de confiabilidad"[5] y seleccionamos un modelo previamente entrenado, como Gemini o ChatGPT, para hacer las predicciones. Los resultados obtenidos se evalúan con las métricas establecidas para medir la eficacia del sistema. En el capítulo siguiente, proporcionaremos una explicación detallada de esta metodología y del desarrollo completo del proyecto.

Existe un enfoque similar al nuestro en el procedimiento de detección de noticias falsas; sin embargo, nuestro método se distingue por utilizar modelos generativos como Gemini y ChatGPT empleando la ingeniería de prompt, lo que

aporta un valor adicional y una perspectiva única en comparación con los estudios previos. Dicho enfoque, *"Fake News Detection on Social Media: A Data Mining Perspective" [19],* ofrece un marco integral para abordar el problema de las noticias falsas mitigando la propagación de desinformación en plataformas digitales. Este sistema de clasificación plantea las siguientes etapas:

En primer lugar, se **define el problema** y **se recopilan los datos**. Los autores del estudio subrayan el desafío clave de distinguir noticias falsas de las verídicas en redes sociales, dada su rápida propagación y el impacto negativo potencial. Esta problemática resalta la necesidad urgente de contar con conjuntos de datos exhaustivos que incluyan tanto ejemplos verídicos como falsos. Estas noticias se clasificaron siguiendo la siguiente función: $F : E \rightarrow \{0, 1\}$

$$F(\alpha) = \{1, \ si\ \alpha\ es\ un\ subtexto\ de\ una\ noticia\ falsa\ \ 0, \ en\ otro\ caso$$

Donde, F = función de predicción que queremos aprender.

$F(\alpha)$ = 1, Indica que el artículo α es considerado una pieza de noticias falsas.

$F(\alpha)$ = 0, Indica que el artículo α no es considerado una pieza de noticias falsas.

Para abordar la detección de noticias se proponen dos fases: la fase de extracción de características y la fase de construcción del sistema.

1. En la **fase de extracción de características,** se identifican y analizan minuciosamente aspectos lingüísticos específicos, así como los atributos asociados a las noticias (fuente, titular, cuerpo de texto, imagen o vídeo) y características de contexto social (usuarios, publicaciones generadas y redes). Estos elementos clave son esenciales para discernir entre contenido auténtico y desinformación. La precisión en esta etapa es fundamental para el éxito de los modelos posteriores de clasificación y detección.
2. En la siguiente fase, el **proceso de modelado** se selecciona un modelo de aprendizaje y este clasifica las noticias según dos contextos. El modelo elegido, se centra en los modelos de contenido de noticias y modelo de contexto social.

    2.1. El modelo de contenido de noticias, se define como un primer enfoque que tiene como objetivo utilizar fuentes externas para verificar los

hechos. Y un segundo enfoque cuyo fin es intentar detectar noticias falsas para capturar a los manipuladores en el estilo de redacción del contenido.

2.2. El modelo de contexto social, se clasifica en dos categorías: la primera utiliza las preferencias de los usuarios a partir de contenidos de publicaciones relevantes para inferir la veracidad de los artículos originales. Estas reacciones pueden incluir "me gusta" y otras interacciones similares.

Y, la segunda categoría se basa en la propagación de noticias falsas.

En segundo lugar, **se predice el resultado** utilizando el conjunto de datos recopilado y se **evalúa** el rendimiento mediante métricas estándar como precisión, recall y F1-score. Estas métricas proporcionan una medida objetiva del éxito de los modelos en la detección de noticias falsas en un entorno dinámico y complejo como las redes sociales.

Como resultado de la investigación, se concluye con la efectividad actual de las estrategias implementadas para detectar noticias falsas en redes sociales. Se destacan las limitaciones identificadas y se proponen recomendaciones específicas para futuros desarrollos tecnológicos y estudios adicionales en este campo crítico para la integridad informativa y la confianza pública.

2.3. Fiabilidad del lenguaje en noticias

Hoy en día, intercambiamos información, fotos o noticias entre nosotros, incluso compartimos contenidos sin entender completamente su significado. Durante esta difusión, la información puede tergiversar.

Como ocurre en la vida real con la difusión de noticias, algunas de estas son creadas por usuarios de redes sociales no identificados, por organizaciones o grupos maliciosos y pueden incluir información ficticia, para confundir a los lectores o receptores de la información. Si estos bulos son intencionalmente difundidos con un

propósito específico, pueden inundar las redes sociales e Internet con información engañosa.

Sin embargo, lo que es conocimiento para los seres humanos no lo es para las máquinas. Estas almacenan datos y estarán a disposición de comenzar el proceso de aprendizaje llevado a cabo por un algoritmo que tratará de analizar y explorar los datos en búsqueda de patrones ocultos.

La gran cantidad de datos disponibles en la web a menudo incluye información incorrecta o engañosa, lo que complica el proceso de verificación de hechos. Además, los algoritmos de los motores de búsqueda no siempre priorizan la veracidad de la información, sino que pueden enfocarse en la relevancia o popularidad del contenido. Esto puede resultar en la difusión de noticias falsas o imprecisas, dificultando aún más la tarea de identificar fuentes confiables y precisas. Por lo tanto, es necesario desarrollar herramientas y técnicas más avanzadas para mejorar la precisión en la búsqueda y evaluación de la información.

La identificación de noticias falsas varía según el idioma debido a las reglas gramaticales y lingüísticas específicas de cada uno. Algunos medios ampliamente reconocidos por difundir información no fiable incluyen plataformas como Twitter, Instagram y Facebook, donde la verificación de hechos puede ser información no fiable. Además, ciertos periódicos y sitios web de noticias, que no siempre siguen estándares estrictos de periodismo, también pueden contribuir a la propagación de noticias falsas.

La tarea de este proyecto es evaluar la fiabilidad del lenguaje en la redacción de noticias en medios digitales. Identificar secciones específicas de un artículo para medir su credibilidad lingüística permite una comprensión más detallada de la veracidad del mensaje y desarrollar técnicas para detectar noticias falsas.

Investigaciones recientes destacan la importancia de examinar estilo, tono y estructura del lenguaje para identificar contenido engañoso. Tanto el estilo como el lenguaje son elementos lingüísticos importantes que ayudan a diferenciar entre artículos reales y falsos, detectando sesgos o distorsiones en el contenido en línea.

2.3.1. "Fake news"

En el estudio anterior, "*Fake News Detection on Social Media: A Data Mining Perspective*" *[19],* el término "fake news" se define como información publicada con la intención de engañar y confundir al lector, imitando la forma de medios fiables, pero ignorando la verificación o adulterando el contenido.

Históricamente, las noticias falsas fueron concebidas para aumentar las ventas de periódicos o para generar miedo o enfado en la población. Sin embargo, hoy por hoy existen innumerables razones para causar desinformación entre la población, siendo la política, social o financiera las motivaciones principales.

Actualmente, las noticias falsas tienen un papel destacado en las redes sociales y afectan significativamente la opinión pública y el ámbito político. Son la principal causa de lo que se conoce como posverdad, donde los hechos objetivos tienen menos influencia que los llamamientos emocionales y las creencias personales. Este término ganó notoriedad en 2016 durante las elecciones presidenciales en Estados Unidos, que resultaron en la elección de Donald Trump, y el referéndum del Brexit, donde se hizo evidente la conexión entre la posverdad y las noticias falsas en la opinión pública.

La IA, y en particular el aprendizaje automático, es clave para identificar y detener la propagación de noticias falsas. Existen numerosas herramientas como las de *AyGLOO[20]* y *ClaimBuster[21]* que utilizan la IA para ayudar a los periodistas a verificar la información, mejorando la transparencia y objetividad de los algoritmos. Además, es esencial que los ciudadanos desarrollen habilidades críticas para evaluar la fiabilidad de las fuentes y reconozcan los sesgos cognitivos que pueden influir en su percepción de la verdad, reduciendo así el impacto de la desinformación.

Las fake news son un problema persistente que requiere un enfoque integral y transversal para su mitigación. Esto implica la colaboración de ciudadanos, gobiernos y organizaciones tecnológicas en la lucha contra la desinformación. A través de la educación, el uso de tecnologías avanzadas y la promoción de un consumo crítico de información, es posible reducir el impacto negativo de estas noticias fraudulentas y proteger los valores democráticos y la convivencia social.

A continuación, se describe la guía RUN-AS, una propuesta realizada por investigadores de la Universidad de Alicante para la clasificación de noticias en las tres posibles categorías (“confiable”, “semiconfiable” o “no confiable”) dependiendo de su fiabilidad.

2.3.2. Guía RUN-AS

RUN-AS (Reliable and Unreliable Annotation Scheme)[7] es un esquema de anotación detallado que categoriza partes estructurales y elementos de contenido esenciales de noticias en "fiables" y "no fiables". Este enfoque ha sido desarrollado por los investigadores: Alba Bonet Jover, Robiert Sepúlveda Torres, Estela Saquete Boro, Patricio Martínez Barco y Mario Nieto Pérez. Su objetivo es identificar patrones de desinformación en el texto y evaluar la fiabilidad global de las noticias.

Para probar la efectividad del enfoque propuesto, se construyó un conjunto de datos en español que fue anotado manualmente utilizando la guía RUN-AS y se realizaron varios experimentos utilizando este conjunto de datos para validar el esquema de anotación mediante algoritmos de Machine Learning (ML) y Deep Learning (DL).

Los resultados del esquema propuesto fueron muy positivos, alcanzando un F1-score máximo de 0,948 con el algoritmo de árbol de decisión. Esta guía se divide en dos apartados a destacar**: tipos de etiquetas** y **criterios de fiabilidad.**

Para los **tipos de etiquetas**, se propone etiquetar noticias en tres niveles: Estructura, Contenido y Elementos de interés, proporcionando ejemplos para ilustrar su aplicación práctica y facilitar la comprensión de cada etiqueta. La siguiente imagen presenta un esquema de anotación que etiqueta estos niveles e incluye atributos clave para mejorar el proceso de evaluación.

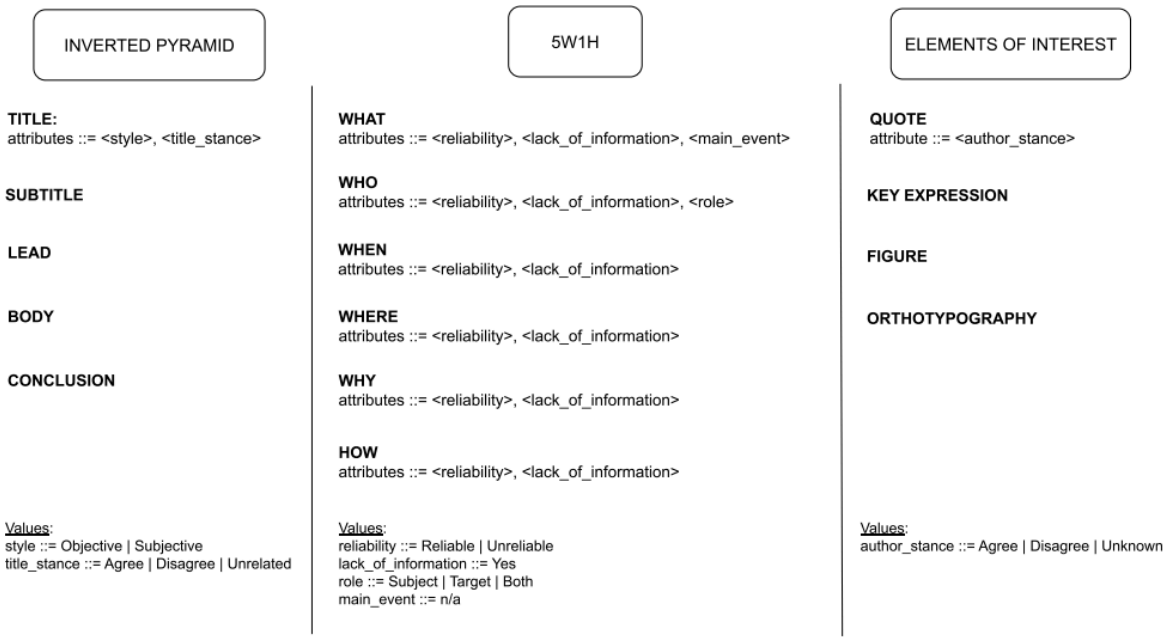

Ilustración 2.2 RUN-AS Guideline
Fuente: Guía RUN-AS[7]

El esquema RUN-AS se fundamenta en principios periodísticos como la pirámide invertida y el enfoque 5W1H. La estructura piramidal invertida organiza la información colocando lo más relevante al principio utilizando las siguientes etiquetas:

- **TITLE**: frase que contiene la idea principal de la noticia y resume la información esencial de una noticia.
- **SUBTITLE**: frase que proporciona información adicional al título.
- **LEAD**: párrafo principal que presenta la información esencial de la noticia respondiendo a las seis preguntas clave del concepto 5W1H.
- **BODY**: conjunto de párrafos que desarrollan la noticia y presentan detalladamente toda la información principal y adicional relacionada con 5W1H.
- **CONCLUSION**: última frase o párrafo que resume el contenido de la noticia. No presenta información nueva y no siempre está presente.

Esto permite una rápida comprensión y cumplimiento de estándares periodísticos. Por otro lado, el método 5W1H asegura que las noticias respondan a

las preguntas clave de quién, qué, cuándo, dónde, por qué y cómo, garantizando una cobertura completa de la información.

Además de estructura y contenido, RUN-AS etiqueta los elementos de interés que ayudan a distinguir noticias fiables de no fiables mediante la identificación de marcadores clave y errores gramaticales mediante las siguientes etiquetas:

- KEY_EXPRESSION: referido a las frases que animan a los lectores a compartir información o expresan emociones como miedo, desprecio, alarma o con fines económicos. Ejemplo: “Vamos a salvar vidas compartiendo esta gran información.” Aquí “vamos a salvar vidas” es la información importante.
- FIGURE: característica que se puede verificar mediante técnicas de verificación de hechos (fact-checking). Por ejemplo, “15 pacientes han dado positivo” es una afirmación que se puede comprobar para verificar su exactitud.
- QUOTE: marca la presencia de citas textuales en la noticia añadiendo autenticidad al artículo. Ejemplo: “El experto dice que es solo <QUOTE> “cuestión de tiempo” <QUOTE>”.
- ORTHOTYPOGRAPHY: cuando la redacción es deficiente y el texto contiene errores gramaticales, ortográficos o de formato.

Este esquema de anotación incluye varios atributos para las etiquetas mencionadas con valores específicos que ayudan a proporcionar información esencial en la anotación:

- **En el método 5W1H:** el atributo principal es la fiabilidad (“**reliability**”), que clasifica cada elemento y la noticia como "Fiable" o "No fiable", considerando la precisión, objetividad y características lingüísticas. Además, el atributo "**main_event**" se utiliza en la etiqueta WHAT para identificar la idea principal y otros elementos relevantes para entender completamente el contexto de la noticia. El atributo "**role**", utilizado con la etiqueta ‘WHO’, especifica el papel desempeñado por la entidad en el evento, como Sujeto, Destino o Ambos. Por otro lado, "**lack_of_information**" señala la falta de pruebas científicas o datos esenciales en las etiquetas 5W1H con el valor "Sí".

- **Para la pirámide invertida:** en la anotación del TÍTULO, contiene la etiqueta "**title_stance**" que evalúa la coherencia con el CUERPO de la noticia con valores como Agree, Disagree o Unrelated. Además, el atributo "**style**" en el TÍTULO indica si la información es objetiva o subjetiva.

- **En los elementos de interés**: destaca el atributo "**author_stance**" con la etiqueta QUOTE que anota la postura del autor con valores como Disagree, Agree o Unknown.

El segundo apartado que presenta la guía trata de **los criterios de fiabilidad**. A pesar de la subjetividad, hay rasgos textuales y lingüísticos que permiten detectar la fiabilidad de una noticia. Se evalúan elementos fiables y no fiables de cada parte de la noticia para valorar la fiabilidad global. Esta guía se ha basado en investigaciones similares a las utilizadas en este trabajo como "*Definir y anotar indicadores de credibilidad en artículos periodísticos*" de Zhang[22]. Los criterios considerados incluyen:

- **Exactitud**: La información debe ser precisa y sin ambigüedades para ser considerada fiable. Las expresiones vagas o imprecisas pueden indicar que se oculta información o que los hechos no están bien fundamentados, lo que disminuye la fiabilidad. Por ejemplo, una fecha exacta o detalles específicos sobre una persona son más fiables que datos generales o inexactos.

- **Objetividad**: La neutralidad es crucial para una noticia fiable. La información debe presentarse de manera objetiva, sin influir en el lector ni mostrar la postura del autor. Los elementos de interés mencionados anteriormente proporcionan pistas sobre la objetividad o subjetividad del contenido.

- **Título**: Los títulos alarmistas o subjetivos suelen indicar noticias menos fiables. En esta anotación, los títulos se etiquetan como objetivos o subjetivos. Los títulos sensacionalistas o que usan muchas mayúsculas y signos de exclamación suelen ser menos confiables.

- **Comentarios personales:** Los comentarios en primera persona o las experiencias personales pueden disminuir la fiabilidad de una noticia, ya que pueden tener la intención de persuadir o manipular al lector.

- **Falta de información:** La ausencia de datos importantes o pruebas, como estudios científicos o datos verificables, puede señalar una falta de fiabilidad. Las noticias que carecen de información esencial o citan estudios sin especificar cuáles, suelen ser menos creíbles.

En esta guía, también se incluyen contenidos adicionales detallados pero no son relevantes ni útiles para el propósito específico de este trabajo.

2.4. Campaña de evaluación IberLEF

Para realizar este proyecto se participará en la campaña IberLEF (Iberian Language Evaluation Forum)[23], una iniciativa de evaluación compartida de sistemas de PLN en español y otras lenguas como portugués, vasco, gallego o catalán vigente desde 2019.

Una campaña de evaluación compartida es una estrategia que busca fomentar la colaboración y la participación de múltiples partes interesadas en la evaluación de un proyecto. En lugar de que una única organización realice la evaluación de manera independiente, se promueve la colaboración y la responsabilidad compartida entre todos los participantes.

Esta campaña en concreto, tiene como objetivo promover la investigación en tareas de procesamiento, comprensión y generación de textos en al menos una de las lenguas nombradas anteriormente.

IberLEF 2024 se celebrará el 24 de septiembre de 2024 en Valladolid en el marco del XL Congreso Internacional de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN 2024) [24]. La SEPLN es una organización sin ánimo de lucro fundada en 1983, promueve el avance del PLN en español mediante conferencias, publicaciones y un congreso anual. También organiza campañas de evaluación, como IberLEF, para involucrar a la comunidad científica en tareas clave del PLN.

Este año se han aceptado doce tareas compartidas para IberLEF 2024, de un total de quince propuestas[25]. Se trata de tareas relacionadas con PLN sobre

identificación de textos generados automáticamente, análisis de contenido clínico y cambio de código, predicción temprana de riesgos en Internet, detección de contenidos nocivos e inclusivos, identificación de ideología política y propaganda, y análisis de sentimientos, posturas y opiniones.

En esta campaña de evaluación compartida, la comunidad investigadora define nuevos retos de investigación y propone tareas para avanzar en el estado del arte del PLN. Estas tareas son revisadas por los miembros de los comités de dirección y programa de IberLEF, y finalmente evaluadas por los presidentes generales de IberLEF.

Los organizadores de las tareas aceptadas, establecen la evaluación de acuerdo con la propuesta presentada, promueven la tarea y gestionan el envío y la evaluación científica de los trabajos de descripción del sistema presentados por los participantes. Estos trabajos científicos se incluyen en las actas del IberLEF y finalmente en la revista de la SEPLN.

Finalmente, los organizadores de las tareas informan de los resultados de las mismas y los participantes seleccionados presentan la descripción de sus sistemas en el taller de IberLEF.

2.5. Tarea FLARES: Clasificación de confiabilidad

Para desarrollar este proyecto se participará en la tarea FLARES: *Fine-Grained Language-Based Reliability Detection in Spanish News* [5], del workshop IberLEF 2024, que tiene como objetivo la detección precisa de la fiabilidad de noticias escritas en español basándose en el lenguaje usado en ellas.

FLARES describe dos subtareas que pueden realizarse de forma independiente para participar en la competición. La primera subtarea, “Identificación de preguntas 5W1H”, requiere que los participantes reciban un texto y determinen su contenido esencial respondiendo a las preguntas 5W1H relacionadas con el documento. En la segunda subtarea, “Confiabilidad basada en 5W1H”, los participantes deberán determinar la fiabilidad de cada ítem 5W1H detectado

clasificándolo como “confiable”, “semiconfiable” y “no confiable”. En este TFG se llevará a cabo el desarrollo de la segunda subtarea.

En el contexto de los medios digitales actuales, evaluar la fiabilidad del lenguaje utilizado en las noticias es cada vez más esencial para determinar la veracidad del contenido. Analizar partes específicas de un artículo, considerando aspectos como el estilo, el tono y la estructura, ayuda a identificar posibles engaños y a desarrollar mejores métodos para detectar noticias falsas. Estudios recientes como “*Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news*”[26] resaltan que el análisis lingüístico es crucial para diferenciar entre noticias verdaderas y falsas y para detectar sesgos en el contenido, consolidándose como una herramienta vital para asegurar la confiabilidad de la información en la era digital.

En esta tarea se propone utilizar la técnica periodística "5W1H" para presentar, de manera clara y precisa, la información clave de una noticia, centrándonos en los elementos Qué, Quién, Por qué, Cuándo, Dónde y Cómo. Aplicar esta técnica permite evaluar sistemáticamente la fiabilidad del lenguaje en estas dimensiones, proporcionando un enfoque estructurado para medir la integridad lingüística y detectar posibles sesgos. El reto de esta tarea utilizará textos en español, con el fin de desarrollar métodos adaptados a este idioma. La integración de esta metodología periodística con el análisis lingüístico ofrece un marco integral que puede mejorar la autenticidad y la confiabilidad de la información en los medios digitales en español.

2.5.1. Corpus

Para participar en la segunda subtarea se proporcionan tres conjuntos de datos distintos, todos ellos en formato JSON, a través de *Kaggle*[27]. Esta plataforma en línea permite a científicos de datos y entusiastas del aprendizaje automático colaborar, encontrar y compartir conjuntos de datos, usar notebooks con GPU y participar en competiciones de ciencia de datos.

El conjunto de datos fue creado por tres expertos altamente cualificados, dos lingüistas y un sociólogo, todos con una sólida formación en PLN, quienes aplicaron

una pauta de anotación precisa y consensuada, la guía RUN-AS[7]. Este dataset está compuesto por aproximadamente 9.034 anotaciones 5W1H, distribuidas a lo largo de 190 noticias recopiladas de forma manual y mediante rastreo web para obtener una variedad representativa de fuentes. Desde los medios más internacionales ABC, BBC News y CNN Español, hasta medios regionales como El Espectador, El Financiero y La Vanguardia, entre otros, se examinaron para garantizar la diversidad de contenido. Rigurosos filtros fueron aplicados para asegurar que cada noticia cumpliera con los estándares lingüísticos, de formato y extensión necesarios para su inclusión en el corpus.

Los tres conjuntos de datos se han proporcionado mediante un enlace de GitHub[28] para su descarga en formato JSON. Estos conjuntos son: el conjunto de datos de entrenamiento o archivo "*5w1h_subtask_2_train.json*", el conjunto de datos de prueba llamado "*5w1h_subtask_2_trial.json*" y el conjunto de datos de test denominado "*5w1h_subtask_2_test.json*".

A continuación, se presenta una tabla y un gráfico de columnas que ilustran la distribución de los datos según la etiqueta de fiabilidad (*Reliability_Label*) en los tres conjuntos de datos, clasificados en las categorías "confiable", "semiconfiable" y "no confiable". Esta tabla facilitará el análisis del rendimiento del modelo en cada una de ellas. Es importante indicar que el conjunto de test no se ha liberado y por ello no se pueden indicar las categorías de este dataset.

	confiable	**semiconfiable**	**no confiable**	**TOTAL**
datos de entrenamiento (5w1h_subtask_2_train.json)	4.765	1.276	893	6.934
datos de prueba (5w1h_subtask_2_trial.json)	435	94	22	551
datos de test (5w1h_subtask_2_test.json)	-	-	-	1.211
TOTAL				**8.696**

Tabla 2.1 Distribución de datos que contienen la etiqueta de fiabilidad

Podemos observar en el gráfico de columnas que los datos en los archivos están desbalanceados. El conjunto de datos de entrenamiento (*train*) supera

ampliamente al conjunto de prueba (*trial*) en todas las categorías, con 4.765 etiquetas frente a las 435 en la categoría confiable 1.276 frente a 94 en semiconfiable y 893 frente a 22 en no confiable. Esta distribución desigual es crucial a la hora de entrenar y evaluar el modelo, ya que el predominio de una categoría podría llevar a un sesgo en las predicciones, favoreciendo a la clase dominante sobre las otras.

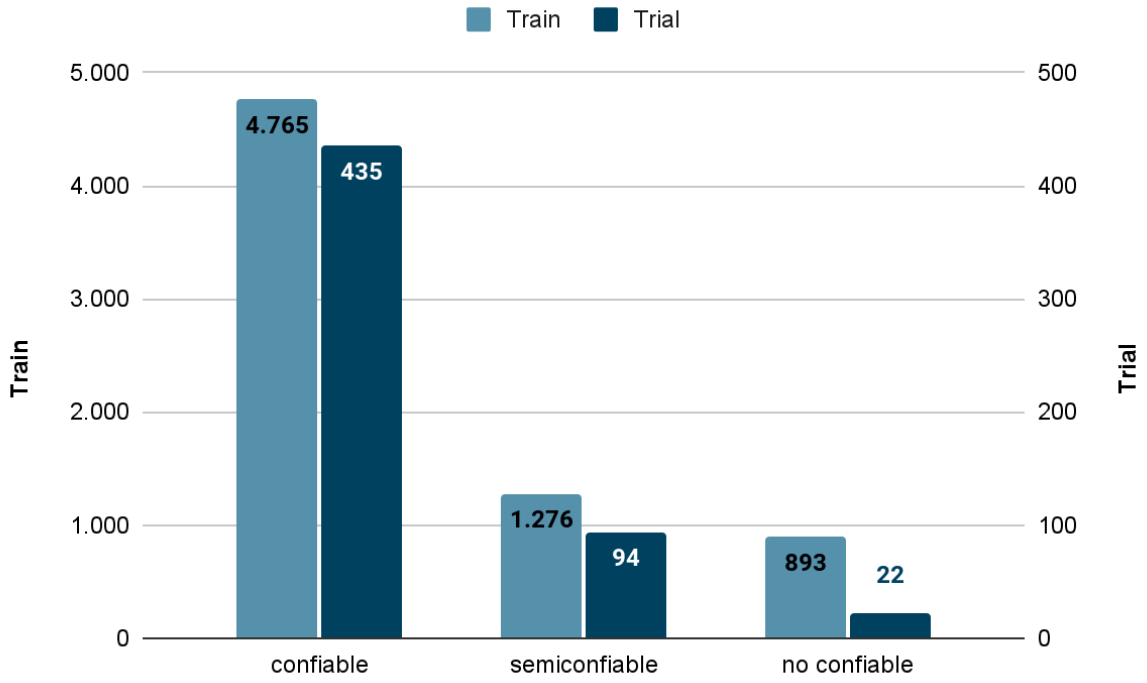

Ilustración 2.3 Distribución de etiquetas de fiabilidad en archivos

Para optimizar la utilidad de los conjuntos de datos, se dividió en dos partes: el 70% (6,934 anotaciones) destinado al entrenamiento y el 30% restante (2,100 anotaciones) reservado para pruebas. Cada anotación, escrita en español, representa una exploración minuciosa de las entidades relevantes en el texto, abordando preguntas fundamentales como Qué, Quién, Dónde, Cuándo, Por qué y Cómo.

La elección del español como idioma principal no solo se fundamenta en su prominencia como el cuarto idioma más hablado en el mundo, sino también en su vital importancia como segunda lengua materna más hablada. Sin embargo es crucial destacar que, a pesar de su relevancia global, la disponibilidad de conjuntos

de datos en español para tareas avanzadas de PLN, como la detección automática de confiabilidad, está claramente limitada.

Este corpus abarca una amplia gama de temas, que van desde la economía hasta la política y la seguridad, reflejando la diversidad y complejidad del panorama informativo contemporáneo. Cada noticia seleccionada, meticulosamente anotada y filtrada, contribuye a la riqueza y la amplitud de este recurso, proporcionando un valioso conjunto de datos para la investigación y el desarrollo en el campo del PLN.

2.5.2. Medidas de evaluación

FLARES propone utilizar las métricas de evaluación propuestas por Piad-Moffis [29] para medir el desempeño en la subtarea 2 (Clasificación de confiabilidad). Estas métricas de evaluación son indispensables para medir el rendimiento de los sistemas, ya que permiten obtener una valoración cuantitativa de su capacidad para realizar predicciones correctas y coherentes. A través de estas medidas, podemos determinar la precisión de las predicciones como el equilibrio entre las categorías correctas e incorrectas [30].

Antes de analizar las métricas de evaluación, es clave comprender algunos términos básicos relacionados con los resultados de un modelo de clasificación. Los "verdaderos positivos" (TP) son los casos en los que el algoritmo predice correctamente una instancia positiva. Los "verdaderos negativos" (TN) son aquellos en los que predice correctamente una instancia negativa. Los "falsos positivos" (FP) ocurren cuando el algoritmo clasifica erróneamente como positivo cuando en realidad es negativo, y los "falsos negativos" (FN) suceden cuando clasifica incorrectamente una instancia positiva como negativa. Estos cuatro resultados forman la base para calcular métricas clave como precisión, cobertura y F1-Score.

- **Accuracy (exactitud)**: es una métrica que mide la proporción de predicciones correctas con respecto al total de predicciones realizadas, por lo que su utilidad se ve muy reducida en casos como esta en la que las clases están desbalanceadas. Sus valores también oscilan entre 0 y 1. Siendo 1 un clasificador que no falla nunca o perfecto.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Donde, *Accuracy = métrica exactitud*

TP = verdaderos positivos

TN = verdaderos negativos

FP = falsos positivos

FN = falsos negativos

- **Precision (precisión)**: es una medida de la exactitud de las predicciones positivas hechas por un clasificador. Se define como la división de verdaderos positivos entre el total de casos clasificados como positivos. Es una métrica muy útil para evaluar la exactitud de las predicciones positivas de un modelo en aplicaciones donde los falsos positivos tienen consecuencias significativas.

$$Precision = \frac{TP}{(TP + FP)}$$

Donde, *Precision = métrica de precisión*

TP = verdaderos positivos

FP = falsos positivos

- **Recall (cobertura)**: es una medida de la capacidad de un modelo de clasificación para identificar todos los ejemplos positivos. Por lo tanto se define como la proporción de verdaderos positivos entre el total de ejemplos que realmente son positivos. Esta métrica es muy útil para cuando es esencial detectar todos los casos positivos aunque haya mayor cantidad de falsos.

$$Recall = \frac{TP}{(TP + FN)}$$

Donde, *Recall = métrica Recall*

TP = verdaderos positivos

FN = falsos negativos

- **F1-Score**: El F1-Score es una métrica comúnmente utilizada en algoritmos de IA para evaluar la capacidad de clasificación. Combina precisión (capacidad de identificar correctamente las instancias positivas) y recuerdo (capacidad de identificar todas las instancias positivas). Sus valores oscilan entre 0 y 1, donde 1 indica un clasificador perfecto. Es especialmente útil para conjuntos de datos desbalanceados, ya que equilibra ambos aspectos en su evaluación.

$$F1 = 2\,x\frac{Precision\ x\ Recall}{(Precision + Recall)}$$

Donde, *F1 = métrica F1-score*

Precision = métrica precisión

Recall = métrica recall o cobertura

3. DESARROLLO

El proyecto se centra en desarrollar un sistema de clasificación automática para determinar la fiabilidad de noticias escritas en español, para ello hemos utilizado los modelos generativos Gemini y ChatGPT. Esta iniciativa está inspirada en los requisitos de la competición “FLARES: Clasificación de fiabilidad” descritos en *Kaggle [27]*. Para su desarrollo, contamos con dos conjuntos de datos: uno para entrenar el modelo y otro para validar su rendimiento. Además, disponemos de un tercer conjunto de datos empleado para probar y verificar la precisión del sistema de clasificación que ha sido entrenado.

Primero, analizaremos los conjuntos de datos utilizados en el desarrollo del sistema, así como el conjunto de datos que facilita la aplicación de las métricas de evaluación de manera efectiva. Posteriormente, se detallará el proceso de implementación del sistema de clasificación automática de fiabilidad, explicando los pasos y las técnicas empleadas para asegurar su correcto funcionamiento. Además, explicaremos las tres técnicas de ingeniería del prompt más utilizadas detallando su aplicación en el proyecto. Se incluirá un ejemplo específico para cada una de estas técnicas que ilustra cómo ha logrado obtener el resultado más óptimo en la evaluación, demostrando su efectividad en la clasificación de fiabilidad.

3.1. Análisis de datos y métricas de evaluación

A continuación, haremos un análisis del conjunto de datos y de las métricas de evaluación proporcionadas por la tarea “FLARES: Clasificación de confiabilidad” que se utilizarán para llevar a cabo la evaluación del sistema.

3.1.1. Análisis del conjunto de datos

Para realizar el análisis del conjunto de datos proporcionados por la competición “FLARES: Clasificación de confiabilidad”, comentada anteriormente, explicaremos: el conjunto de datos de entrenamiento (*5w1h_subtask_2_train.json*), el conjunto de datos de prueba (*5w1h_subtask_2_trial.json*) y el conjunto de datos de test (*5w1h_subtask_2_test.json*).

Cada línea de cada uno de los tres archivos que se han facilitado, representa una estructura de datos llamado diccionario. Estas estructuras almacenan pares clave-valor permitiendo un acceso rápido y eficiente a los datos. Cada línea cuenta con siete pares clave-valor separadas por coma, descritas brevemente a continuación:

- **Id:** identificador único para cada entrada.
- **Text:** titular de la noticia. Cabe destacar que contiene caracteres Unicode (\u00). Estos son una representación universal de un carácter. Por ejemplo, en España, la representación de la letra “ñ” corresponde con el código ‘\u00F1’.
- **Reliability_Label**: proporciona información sobre la fiabilidad de la etiqueta, clasificada como confiable, semiconfiable o no confiable.
- **5W1H_Label:** proporciona una clasificación basada en el método 5W1H (Quién, Qué, Dónde, Cuándo, Por qué y Cómo). Esta etiqueta está estrechamente relacionada con la etiqueta *Tag_Text* ya que este valor dependerá de la porción de texto.
- **Tag_Text:** presenta una porción de texto extraído de la etiqueta Text.
- **Tag_Start:** indica la posición de inicio de la porción de texto destacada en el campo Text, es decir, el carácter dónde empieza el contenido de la etiqueta *Tag_Text.*

- **Tag_End:** indica la posición de fin de la porción de texto destacada en el campo Text, es decir, el carácter dónde termina el contenido de la etiqueta *Tag_Text*.

Estos siete pares de identificadores y valores están presentes únicamente en los conjuntos de entrenamiento y prueba, ya que en el conjunto de test se prescinde de la etiqueta *Reliability_Label*. Para llevar a cabo el objetivo del proyecto, trabajaremos con todas estas etiquetas excepto con las dos últimas.

El conjunto de entrenamiento consta de 6.934 líneas de código, mientras que el conjunto de prueba incluye 551 líneas y el conjunto de test 1.211 líneas. Ahora observaremos una línea de código de cada uno de los archivos:

```Python
// Conjunto de entrenamiento (5w1h_subtask_2_train.json)

{"Id":1811,"Text":"Todos los preparados probados redujeron el virus.
","Reliability_Label":"semiconfiable","5W1H_Label":"WHAT","Tag_Text":"Todos
los preparados probados","Tag_Start":0,"Tag_End":29}
```

```Python
// Conjunto de prueba (5w1h_subtask_2_trial.json)

{"Id":280,"Text":"Los casos se disparan en Reino
Unido","Reliability_Label":"confiable","5W1H_Label":"WHERE","Tag_Text":"en
Reino Unido","Tag_Start":22,"Tag_End":36}
```

```Python
// Conjunto de test (5w1h_subtask_2_test.json)

{"Id":634,"Text":"La mejor dieta del mundo: Adelgazar comiendo jam\u00f3n y
vino","5W1H_Label":"WHAT","Tag_Text":"Adelgazar","Tag_Start":26,"Tag_End":35}
```

Es importante saber que en los tres archivos, el contenido de la etiqueta *Text* se puede repetir varias veces debido a que se hace referencia a una porción de noticia diferente. Para ilustrar mejor el proceso, veamos un ejemplo:

```
Python
{"Id":6917,"Text":"El cerebro del ni\u00f1o, o del adolescente, est\u00e1
sediento de ox\u00edgeno.
","Reliability_Label":"confiable","5W1H_Label":"WHO","Tag_Text":"del
ni\u00f1o, o del adolescente","Tag_Start":11,"Tag_End":38}

{"Id":6918,"Text":"El cerebro del ni\u00f1o, o del adolescente, est\u00e1
sediento de ox\u00edgeno. ","Reliability_Label":"no
confiable","5W1H_Label":"WHAT","Tag_Text":"sediento de
ox\u00edgeno","Tag_Start":45,"Tag_End":64}
```

Dado que las etiquetas *Reliability_Label* y *5W1H_Label* están relacionadas significativamente, será interesante saber cuántas etiquetas pertenecen a cada una de las preguntas *5W1H* tanto en el conjunto de entrenamiento como en el conjunto de prueba. Esta información nos permitirá guiar al prompt con los ejemplos en los que debe fijarse.

	confiable	semiconfiable	no confiable	TOTAL
WHO	1.395	132	316	**1.843**
WHERE	659	40	102	**801**
HOW	267	191	105	**563**
WHEN	571	42	165	**778**
WHAT	1.722	440	549	**2.711**
WHY	151	48	39	**238**

Tabla 3.1 Distribución de etiquetas de fiabilidad y 5W1H del conjunto de entrenamiento

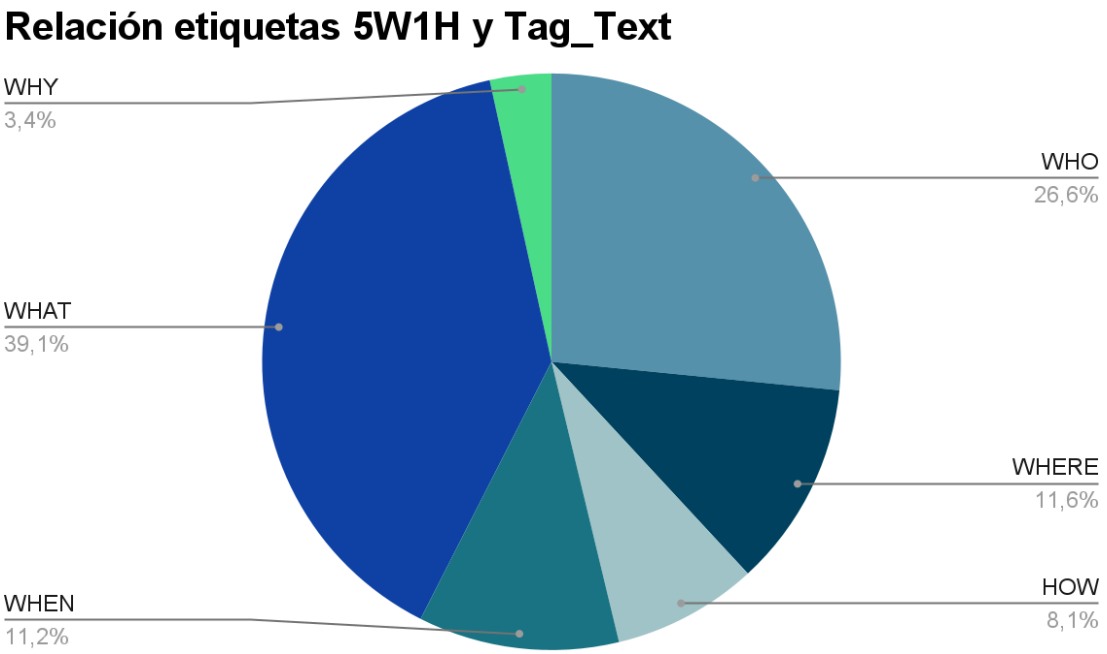

Ilustración 3.2 Distribución de etiquetas 5W1H en relación con el contenido de Tag_Text del conjunto de datos de entrenamiento

	confiable	semiconfiable	no confiable	TOTAL
WHO	116	22	0	**138**
WHERE	57	7	2	**66**
HOW	22	12	6	**40**
WHEN	47	15	0	**62**
WHAT	187	37	14	**238**
WHY	6	1	0	**7**

Tabla 3.2 Distribución de etiquetas de fiabilidad y 5W1H del conjunto de prueba

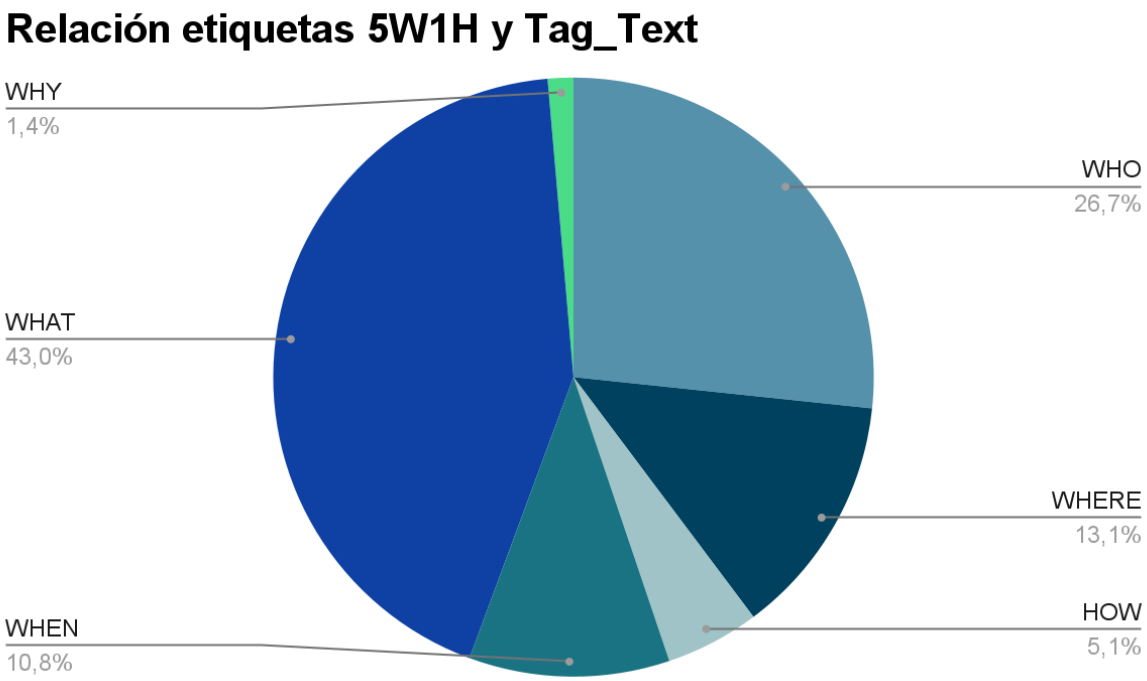

Ilustración 3.3 Distribución de etiquetas 5W1H en relación con el contenido de Tag_Text del conjunto de datos de prueba

Aquí también podemos observar un desequilibrio en los datos. En este caso, esta desproporción es un aspecto importante que debemos considerar al proporcionar ejemplos al sistema en técnicas que los requieren.

Una vez analizado el contenido de los archivos procedo a analizar las métricas de evaluación usadas en la tarea FLARES para evaluar los sistemas.

3.1.2. Análisis de las métricas de evaluación

Ahora, examinaremos cómo los conjuntos de datos se utilizan para evaluar el desempeño del sistema, comparando los datos reales con las predicciones generadas para calcular las métricas de evaluación, como la precisión, el recall, la exactitud y la métrica F1-score. Dado que contamos con un conjunto de datos muy desbalanceado, se elige centrar este trabajo en el F1-Score, ya que es la métrica que ofrece un equilibrio armonioso entre precisión y cobertura, siendo la más adecuada para este problema de clasificación.

Estas cuatro métricas de evaluación, están definidas en los dos archivos siguientes usados en la tarea FLARES para evaluar los sistemas: el primero "*evaluation_metric_template_kaggle_subtask_2*" se centra en la implementación de la métrica de evaluación que será utilizada en la competición de Kaggle y el segundo, "*evaluate_subtask_2*" es un script que nos permite comparar las predicciones con los datos reales y calcular las métricas de rendimiento localmente.

En el primer archivo encontramos un comentario inicial que nos servirá como guía para la documentación de las métricas en Kaggle. Y, posteriormente, se importan las bibliotecas requeridas: *pandas* para la manipulación de data frames y las funciones de *sklearn.metrics* para el cálculo de las métricas de evaluación.

```python
"""
TODO: Enter any documentation that only people updating the metric should
read here.

All columns of the solution and submission dataframes are passed to your
metric, except for the Usage column.

Your metric must satisfy the following constraints:
- You must have a function named score. Kaggle's evaluation system will call
that function.
- You can add your own arguments to score, but you cannot change the first
three (solution, submission, and row_id_column_name).
- All arguments for score must have type annotations.
- score must return a single, finite, non-null float.

"""

import pandas as pd
from sklearn.metrics import precision_score, recall_score, f1_score,
accuracy_score
```

El siguiente bloque de código define una clase de excepción personalizada diseñada para mostrar mensajes de error específicos a los participantes de la competición, mientras que otros errores se gestionan internamente para evitar la filtración de datos privados y proteger la información sensible.

```python
class ParticipantVisibleError(Exception):
```

```
    # If you want an error message to be shown to participants,
       you must raise the error as a ParticipantVisibleError
    # All other errors will only be shown to the competition host.
       This helps prevent unintentional leakage of solution data.
    pass
```

La siguiente función evalúa las predicciones realizadas por el sistema comparándolas con los datos reales. Primero, verifica que los valores de la etiqueta *"Id"* coincidan en ambos ficheros; si no es así, se lanza un error utilizando la clase de excepción personalizada mencionada anteriormente. Luego, extrae las etiquetas reales ("y_test") y las etiquetas predichas ("y_hat"). Utiliza la biblioteca *sklearn* para calcular las métricas de rendimiento, todas ellas con un promedio macro que trata todas las clases de manera equitativa. Finalmente, la función retorna un diccionario con estas métricas calculadas.

Python

```python
def eval_dataset(solution: pd.DataFrame, submission: pd.DataFrame):
    if list(solution['Id']) != list(submission['Id']):
        raise ParticipantVisibleError(f'The ID Column of the solution
                                        and submission do not match')

    y_test = list(solution['Reliability_Label'])
    y_hat = list(submission['Reliability_Label'])

    f1_score_value = f1_score(y_test, y_hat, average='macro')
    accuracy_value = accuracy_score(y_test, y_hat)
    precision_value = precision_score(y_test, y_hat, average='macro')
    recall_value = recall_score(y_test, y_hat, average='macro')
    metrics = {'F1': f1_score_value, 'Accuracy': accuracy_value,
             'Precision': precision_value, 'Recall': recall_value}

    return metrics
```

Por último, se define la función principal, encargada de calcular la métrica F1-score para evaluar el rendimiento del sistema. Esta función invoca a "eval_dataset" para calcular las métricas de rendimiento, elimina la columna "Id" de los data frames, ya que Kaggle solo la utiliza para alinear los datos, y finalmente devuelve el valor del F1-score con un promedio macro. Este valor será clave para determinar la calidad de las predicciones en la competición.

```python
def score(solution: pd.DataFrame, submission: pd.DataFrame,
row_id_column_name: str) -> float:
    '''
    TODO: Document your metric here. This docstring will be shown when
          selecting the metric on Kaggle.
    The docstring must be fewer than 8,000 characters long.

    Accuracy, Precision, Recall, and F1 metrics calculated using the sklearn
          library.
    Precision, Recall, and F1 are calculated using the macro average method.
    Return F1 macro score
    '''
    metrics = eval_dataset(solution, submission)

    # Delete the row ID column, which Kaggle's system uses to align the
       solution and submission before passing these dataframes to score().
    del solution[row_id_column_name]
    del submission[row_id_column_name]

    #Return the metric
    return metrics['F1']
```

El segundo archivo primero se importan las librerías necesarias para el script, que incluyen *argparse* para gestionar los argumentos de línea de comandos, *os* para manejar rutas de archivos y *pandas* como hemos comentado anteriormente, para gestionar dataframes. También importamos el primer archivo para evaluar las predicciones “eval_dataset”. Además, añade un bloque de comentario que proporciona un ejemplo del formato esperado para el archivo de predicciones que se evaluará, indicando que debe contener dos columnas: “Id” y “Reliability_Label”.

```python
import argparse
import os
import pandas as pd
from evaluation_metric_template_kaggle_subtask_2 import eval_dataset

"""
    Submission format:
    Id,Reliability_Label
    75,confiable
    179,confiable
    511,confiable
```

```
"""
```

La función main se encarga de cargar los datos reales *dataGold* y las predicciones *dataInfered* desde los archivos CSV cuyas rutas se especifican en los argumentos de línea de comandos. Luego utiliza la función “eval_dataset” para evaluar las predicciones y calcular las métricas correspondientes. Finalmente, imprime los resultados de las métricas obtenidas.

```python
def main(args):
    pathDataGold = args.pathDataGold
    pathDataInfered = args.pathDataInfered
    dataGold = pd.read_csv(os.getcwd() + pathDataGold, sep=",")
    dataInfered = pd.read_csv(os.getcwd() + pathDataInfered, sep=",")

    metrics = eval_dataset(dataGold, dataInfered)
    print(metrics)
```

El último bloque de código facilita la ejecución del script desde la línea de comandos. Configura *ArgumentParser* para gestionar los argumentos de entrada, permitiendo especificar las rutas de los archivos de datos reales *pathDataGold* y las predicciones *pathDataInfered*. Si no se proporcionan estas rutas, se utilizan valores predeterminados. Para finalizar, llama a la función main() para ejecutar la lógica principal del script con los argumentos proporcionados.

```python
if __name__ == "__main__":
    parser = argparse.ArgumentParser(description='Process some integers.')
    parser.add_argument('--pathDataGold', metavar='pathDataGold', type=str,
                        help='Path to the data real',
                        default='/data/gold.csv')
    parser.add_argument('--pathDataInfered', metavar='pathDataInfered',
                        type=str, help='Path to the data infered',
                        default='/data/submission.csv')

    args = parser.parse_args()
    main(args)
```

Estos dos conjuntos de archivos, no solo permiten validar la precisión y eficacia de nuestro sistema sino que también proporcionan una estructura clara y robusta para su evaluación automatizada. Trabajan juntos para proporcionar un sistema de evaluación completo: uno permite la evaluación local de las predicciones, y el otro asegura que la métrica personalizada se integre correctamente con la plataforma de Kaggle. Este enfoque garantiza que las predicciones de nuestro sistema sean evaluadas de manera precisa y consistente, tanto durante el desarrollo local como en el entorno competitivo de Kaggle.

Tras analizar los cinco conjuntos de datos proporcionados en la tarea "FLARES: Clasificación de confiabilidad" [5] ("*5w1h_subtask_2_train.json", "5w1h_subtask_2_trial.json", "5w1h_subtask_2_test.json", "evaluate_subtask_2" y "evaluation_metric_template_kaggle_subtask_2"*), pasamos a explicar el sistema de clasificación diseñado para la detección de fiabilidad en noticias en español.

3.2. Sistema de clasificación para la detección de fiabilidad

Antes de explicar el sistema y su funcionamiento, es importante mencionar que el sistema se ha desarrollado para participar en la tarea FLARES, organizada en IberLEF por los investigadores: Estela Saquete Boro, Alba Bonet Jover, Robiert Sepúlveda Torres, Patricio Martínez Barco, Carlos Badenes Olmedo, Ibai Guillén Pacho, Isam Diab Lozano, Isabel Cabrera de Castro, María Teresa Martín Valdivia y Luis Alfonso Ureña López. Estos organizadores son parte de grupos de investigación especializados en PLN y Sistemas de Información de distintas universidades incluyendo la Universidad de Alicante, la Universidad Politécnica de Madrid y la Universidad de Jaén [31].

Para implementar un sistema de clasificación efectivo utilizando prompts, se utilizará la guía de ingeniería de prompts[14]. Estas guías proporcionan un marco estructurado que facilita la creación y optimización de prompts, asegurando que el sistema pueda interpretar y responder adecuadamente a una variedad de entradas.

La ingeniería de prompts es crucial para mejorar la precisión y relevancia de las respuestas generadas, lo que, a su vez, aumenta la eficiencia del sistema de

clasificación. Además, al seguir estas directrices, se garantiza que el sistema sea adaptable y escalable, permitiendo futuras mejoras y ajustes según sea necesario.

El objetivo de este proyecto es determinar la fiabilidad del subtexto de la noticia para cada uno de los elementos 5W1H detectados siguiendo la pauta RUN-AS[7] (*Reliable and Unreliable Annotation Scheme*). Esta evaluación utiliza el atributo "confiabilidad", que categoriza el lenguaje como "confiable" o "no confiable" según dos criterios clave. Por un lado, se considera la exactitud del contenido, evaluando aspectos como la precisión en las fechas o referencias específicas. Por ejemplo, frases como "Hace mucho tiempo" se consideran inexactas y por ende menos confiables, en contraste con declaraciones precisas como "El viernes 19 de marzo". Por otro lado, se evalúa la neutralidad del contenido, analizando la presencia de opiniones personales, emociones o sesgos en el texto. Por ejemplo, expresiones subjetivas como "En mi opinión" se catalogan como menos fiables en comparación con declaraciones objetivas como "Según el rector de la Universidad de Alicante".

La guía de RUN-AS únicamente clasifica las etiquetas “confiable” y “no confiable”. El atributo “semiconfiable” se creó posteriormente como grado intermedio a estas dos etiquetas, especialmente como parte de la tarea compartida, por eso no aparece en el artículo. Para definirlas brevemente, la etiqueta "confiable" se utiliza cuando la información es tanto objetiva como precisa; la etiqueta "no confiable" cuando no cumple ni el criterio de objetividad ni el de precisión, y la etiqueta "semiconfiable" se utilizará cuando solo cumpla uno de los dos criterios.

Disponemos de tres archivos de datos que son esenciales para el desarrollo correcto de la tarea. Estos son el conjunto de datos de entrenamiento, el conjunto de datos de pruebas y el conjunto de datos de test. Posteriormente, se realizará un análisis más detallado de estos datos.

Para realizar el funcionamiento del sistema destinado a evaluar la detección de la fiabilidad de las noticias proporcionadas, se aplicarán los siguientes pasos para cada una de las técnicas de prompting utilizadas. En la siguiente imagen se muestra un diagrama de actividad para el flujo de ejecución de cada técnica.

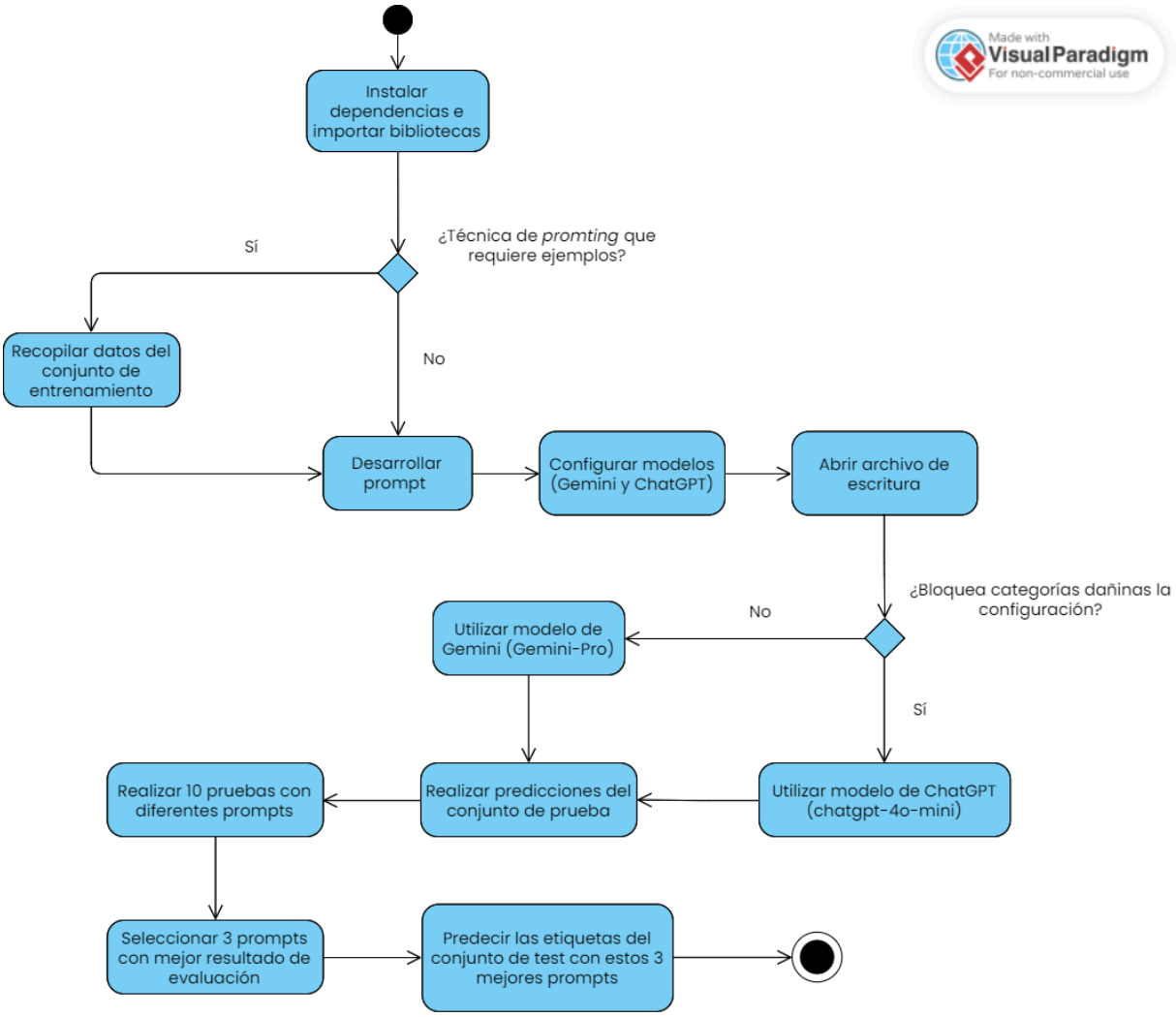

Ilustración 3.1 Diagrama de actividad para el flujo de ejecución de cada técnica.

Fuente: creación propia

1. Instalamos todas las dependencias e importamos las bibliotecas necesarias para que nuestro sistema se pueda ejecutar correctamente.
2. En caso de que utilicemos una técnica de prompting que requiera ejemplos, recopilamos los datos del archivo *train.json*. Estos serán los ejemplos que le proporcionaremos al prompt como referencia.
3. Desarrollamos el prompt describiendo la acción que debe de hacer el sistema para predecir.
4. Configuramos el modelo de Gemini Pro y de Chat-GPT.

5. Abrimos el archivo de escritura para escribir las soluciones que predice el sistema. Primero, verifica si el contenido a evaluar pertenece a una de las etiquetas 5W1H (‘WHO’, ‘WHERE’, ‘HOW’, ‘WHEN’, ‘WHAT’, ‘WHY’). Si es así, intenta utilizar el modelo principal (gemini-pro) para generar la respuesta aplicando configuraciones de seguridad para filtrar contenido sensible. En el caso de que, el contenido actual pertenezca a una de las categorías de contenido dañino o inadecuado (peligroso, acoso, discurso de odio, sexualmente explícito, contenido peligroso) la generación de contenido será bloqueada. Cuando esto ocurre, el sistema recurre al modelo alternativo (gpt-4o-mini) para continuar el proceso de predicción de la fiabilidad de la noticia.

6. La respuesta generada, ya sea por el modelo principal o el alternativo, se almacena en el archivo CSV junto con el identificador correspondiente de cada entrada. Utilizaremos el archivo de prueba (*5w1h_subtask_2_trial.json*), que contiene datos etiquetados con la categoría “Reliability_Label”, para evaluar su comportamiento y comprobar cuál ofrece mejor resultado. Este archivo de prueba incluye ejemplos de textos clasificados en diferentes categorías de fiabilidad, lo que nos permite comparar cómo cada modelo maneja las distintas situaciones.

7. Para evaluar el desempeño de los sistemas se realizarán diez pruebas utilizando distintos prompts. Estos prompts están diseñados para explorar cómo varía la capacidad de los modelos generativos para generar respuestas precisas y coherentes en función de los diferentes enfoques de entrada. Los resultados obtenidos de estas pruebas nos permitirán identificar los tres prompts que muestran el mejor desempeño, facilitando así la selección de los enfoques más efectivos para la generación de contenido fiable.

8. Por último, utilizaremos estos tres prompts seleccionados para realizar las predicciones con el archivo test (*5w1h_subtask_2_test.json*) con el objetivo de predecir todos los valores de nuestro conjunto de datos de test.

A continuación, procederemos a implementar el código del sistema de clasificación de confiabilidad completo de nuestro proyecto integrando todas las

funcionalidades necesarias. Este paso implica reunir y organizar los distintos componentes que se han desarrollado, asegurando que trabajan de manera coherente para lograr los objetivos del proyecto.

Primero, deberemos instalar las bibliotecas necesarias para utilizar los modelos de Gemini y chatGPT.

```Python
!pip install openai
!pip install google.generativeai
```

El primer comando instala la biblioteca *openai*, que permite interactuar con los modelos de lenguaje de OpenAI a través de su API. Esto da acceso al modelo ChatGPT, facilitando la integración de funciones avanzadas de generación de texto y PLN en aplicaciones y proyectos.

La segunda instrucción instala la biblioteca *google.generativeai*, necesaria para trabajar con el modelo de IA generativa de Google, Gemini. Esta biblioteca facilita el uso de capacidades avanzadas de generación de texto y otros servicios proporcionados por Google en proyectos de Python.

Ahora, importamos las bibliotecas necesarias para manejar datos en formato JSON y CSV, lo que nos permitirá leer, escribir y procesar información estructurada dentro de nuestro proyecto.

```Python
import json
import csv
```

El desarrollo del código dependerá de la técnica de prompt que se elija, ya que cada enfoque exige una implementación particular.

En caso de que la técnica requiera ejemplos como en “***few-shot”*** y “**CoT-few-shot”**, clasificamos los datos del archivo de entrada en formato JSON en listas específicas basadas en dos criterios: la categoría *5W1H_Label* (‘WHO’, ‘WHERE’, ‘HOW’, ‘WHEN’, ‘WHAT’, ‘WHY’) y la fiabilidad *Reliability_Label*

(“'confiable”, “semiconfiable” o “no confiable”). Para cada línea del archivo JSON, identificamos la categoría y la fiabilidad, y luego añadimos el objeto JSON correspondiente a la lista adecuada. Se ha decidido colocar el objeto justo antes del último elemento de la lista ya que ha demostrado un mejor rendimiento en comparación con agregarlo al principio o al final.

Python

```
WHOS_CONFIABLES = []
WHOS_SEMICONFIABLES = []
WHOS_NOCONFIABLES = []

WHERE_CONFIABLES = []
WHERE_SEMICONFIABLES = []
WHERE_NOCONFIABLES = []

HOW_CONFIABLES = []
HOW_SEMICONFIABLES = []
HOW_NOCONFIABLES = []

WHEN_CONFIABLES = []
WHEN_SEMICONFIABLES = []
WHEN_NOCONFIABLES = []

WHAT_CONFIABLES = []
WHAT_SEMICONFIABLES = []
WHAT_NOCONFIABLES = []

WHY_CONFIABLES = []
WHY_SEMICONFIABLES = []
WHY_NOCONFIABLES = []

# Archivo de entrada
input_file = 'C:/Users/elena/Desktop/train.json'

with open(input_file, 'r') as file:
for line in file:
    json_object = json.loads(line)

    Text = json_object['Text']
    Tag_Text = json_object['Tag_Text']
    Tag = json_object['5W1H_Label']
    Label = json_object['Reliability_Label']

    if Tag == 'WHO':
        if Label == "confiable":
            WHOS_CONFIABLES.insert(-1, json_object)
```

```
        elif Label == "semiconfiable":
            WHOS_SEMICONFIABLES.insert(-1, json_object)
        elif Label == "no confiable":
            WHOS_NOCONFIABLES.insert(-1, json_object)

    elif Tag == 'WHERE':
        if Label == "confiable":
            WHERE_CONFIABLES.insert(-1, json_object)
        elif Label == "semiconfiable":
            WHERE_SEMICONFIABLES.insert(-1, json_object)
        elif Label == "no confiable":
            WHERE_NOCONFIABLES.insert(-1, json_object)

    elif Tag == 'HOW':
        if Label == "confiable":
            HOW_CONFIABLES.insert(-1, json_object)
        elif Label == "semiconfiable":
            HOW_SEMICONFIABLES.insert(-1, json_object)
        elif Label == "no confiable":
            HOW_NOCONFIABLES.insert(-1, json_object)

    elif Tag == 'WHEN':
        if Label == "confiable":
            WHEN_CONFIABLES.insert(-1, json_object)
        elif Label == "semiconfiable":
            WHEN_SEMICONFIABLES.insert(-1, json_object)
        elif Label == "no confiable":
            WHEN_NOCONFIABLES.insert(-1, json_object)

    elif Tag == 'WHAT':
        if Label == "confiable":
            WHAT_CONFIABLES.insert(-1, json_object)
        elif Label == "semiconfiable":
            WHAT_SEMICONFIABLES.insert(-1, json_object)
        elif Label == "no confiable":
            WHAT_NOCONFIABLES.insert(-1, json_object)

    elif Tag == 'WHY':
        if Label == "confiable":
            WHY_CONFIABLES.insert(-1, json_object)
        elif Label == "semiconfiable":
            WHY_SEMICONFIABLES.insert(-1, json_object)
        elif Label == "no confiable":
            WHY_NOCONFIABLES.insert(-1, json_object)
```

Ahora es momento de redactar el prompt cuya redacción dependerá de la técnica utilizada. En el próximo apartado profundizaremos cada una de estas técnicas en detalle; por ahora, solo presentaremos un ejemplo generalizado:

```python
def getQuery2(text, tag_5w1h, subtext):
    PROMPT_TEMPLATE = """

       ### Instrucciones para analizar y predecir la fiabilidad del texto ###

     """
    PROMPT_TEMPLATE += "\n\nInformación de Entrada:\nTexto: " + text + ". " +
       "5W1H_Label: " + tag_5w1h + ". Subtexto: " + subtext

    return PROMPT_TEMPLATE
```

En el caso de que la técnica requiera ejemplos, debemos agregarlos al prompt usando bloques *if-elif* según su correspondiente etiqueta *5W1H_Label*. Cada bloque se encarga de clasificar los ejemplos del archivo de entrenamiento (*train.json*) y los incorpora al prompt en el formato adecuado. Finalmente, incluimos la información de entrada proporcionada para completar el prompt:

```python
def getQuery2(text, tag_5w1h, subtext):
    PROMPT_TEMPLATE = """

    ### Instrucciones para analizar y predecir la fiabilidad del texto ###

    ### Ejemplos ###

    """
    if tag_5w1h == 'WHO':
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
              WHOS_CONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
              WHOS_CONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
              WHOS_CONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
              WHOS_CONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
              WHOS_SEMICONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
              WHOS_SEMICONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
              WHOS_SEMICONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
              WHOS_SEMICONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
              WHOS_NOCONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
              WHOS_NOCONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
```

```
            WHOS_NOCONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHOS_NOCONFIABLES[x]['Reliability_Label']

    elif tag_5w1h == 'WHERE':
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHERE_CONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHERE_CONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHERE_CONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHERE_CONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHERE_SEMICONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHERE_SEMICONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHERE_SEMICONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHERE_SEMICONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHERE_NOCONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHERE_NOCONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHERE_NOCONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHERE_NOCONFIABLES[x]['Reliability_Label']

    elif tag_5w1h == 'HOW':
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            HOW_CONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            HOW_CONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            HOW_CONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            HOW_CONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            HOW_SEMICONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            HOW_SEMICONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            HOW_SEMICONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            HOW_SEMICONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            HOW_NOCONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            HOW_NOCONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            HOW_NOCONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            HOW_NOCONFIABLES[x]['Reliability_Label']

    elif tag_5w1h == 'WHEN':
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHEN_CONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
```

```
            WHEN_CONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHEN_CONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHEN_CONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHEN_SEMICONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHEN_SEMICONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHEN_SEMICONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHEN_SEMICONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHEN_NOCONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHEN_NOCONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHEN_NOCONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHEN_NOCONFIABLES[x]['Reliability_Label']

    elif tag_5w1h == 'WHAT':
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHAT_CONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHAT_CONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHAT_CONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHAT_CONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHAT_SEMICONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHAT_SEMICONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHAT_SEMICONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHAT_SEMICONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHAT_NOCONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHAT_NOCONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHAT_NOCONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHAT_NOCONFIABLES[x]['Reliability_Label']

    elif tag_5w1h == 'WHY':
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHY_CONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHY_CONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHY_CONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHY_CONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHY_SEMICONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHY_SEMICONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
```

```
            WHY_SEMICONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHY_SEMICONFIABLES[x]['Reliability_Label']
        for x in range(0, 5):
            PROMPT_TEMPLATE += "\n    Texto: " +
            WHY_NOCONFIABLES[x]['Text'] + ". " + "5W1H_Label: " +
            WHY_NOCONFIABLES[x]['5W1H_Label'] + ". Subtexto: " +
            WHY_NOCONFIABLES[x]['Tag_Text'] + ". Respuesta: " +
            WHY_NOCONFIABLES[x]['Reliability_Label']

    PROMPT_TEMPLATE += "\n\nInformación de Entrada:\nTexto: " + text + ". " +
"5W1H_Label: " + tag_5w1h + ". Subtexto: " + subtext

    return PROMPT_TEMPLATE
```

Una vez desarrollado el prompt, procedemos a importar las bibliotecas necesarias para manejar el modelo generativo de IA y configurar la clave API para su uso. Configuramos el modelo ***gemini-pro*** de Google Generative AI con nuestra clave API personal.

Python
```
input_file_test = 'C:/Users/elena/Desktop/trial.json'

# Importamos Gemini Pro
import google.generativeai as genai

# Usamos el modelo generativo de la IA
modelo = genai.GenerativeModel('gemini-pro')

# Configuramos la API KEY
GOOGLE_API_KEY='AIzaSyCOxj-ayI-GoogleGenerativekey'
genai.configure(api_key=GOOGLE_API_KEY)
```

Es importante que configuremos los ajustes de seguridad para filtrar el contenido potencialmente dañino en las respuestas generadas por el modelo de IA, de modo que se eviten excepciones que puedan interrumpir la ejecución.

Python
```
safety_settings = [
    {
        "category": "HARM_CATEGORY_DANGEROUS",
        "threshold": "BLOCK_NONE",
```

```
    },
    {
        "category": "HARM_CATEGORY_HARASSMENT",
        "threshold": "BLOCK_NONE",
    },
    {
        "category": "HARM_CATEGORY_HATE_SPEECH",
        "threshold": "BLOCK_NONE",
    },
    {
        "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
        "threshold": "BLOCK_NONE",
    },
    {
        "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
        "threshold": "BLOCK_NONE",
    },
]
```

Definimos la ruta del archivo CSV donde se guardarán los resultados y configuramos el cliente de OpenAI con nuestra clave API privada.

```python
output_file_csv='C:/Users/elena/Desktop/salida.csv'

from openai import OpenAI
client = OpenAI(
    api_key="sk-proj-openAIkey",
)
```

Por último abrimos el archivo CSV de salida en modo escritura, incluyendo en la cabecera del archivo las columnas 'Id' y 'Reliability_Label'. Se lee línea por línea del archivo JSON de entrada, se parsea cada línea como un objeto JSON y se extraen las claves relevantes ('Id', 'Text', 'Tag_Text', '5W1H_Label'). Se intenta generar una respuesta utilizando el modelo Gemini. Si el contenido resulta ser dañino, utilizamos el modelo ***gpt-4o-mini*** para predecir la fiabilidad de la noticia. Una vez que se obtiene una respuesta válida, se escribe en el archivo CSV junto con el identificador correspondiente del objeto. Esta última parte está relacionada con el post-procesamiento la cual profundizaremos más adelante en el documento.

Python

```
# Abre el archivo CSV para escritura
with open(output_file_csv, 'w', newline='') as csvfile:
    csv_writer = csv.writer(csvfile)
    # Escribe la cabecera del archivo CSV
    csv_writer.writerow(['Id', 'Reliability_Label'])

    with open(input_file_test, 'r') as file:
        for line in file:
            json_object = json.loads(line)
            Id = json_object['Id']
            Text = json_object['Text']
            Tag_Text = json_object['Tag_Text']
            Tag = json_object['5W1H_Label']

            response = None

            while True:
                if Tag in ['WHO', 'WHERE', 'HOW', 'WHEN', 'WHAT', 'WHY']:
                    try:
                        respuesta = modelo.generate_content(
                                 getQuery2(Text, Tag, Tag_Text),
                                 safety_settings=safety_settings)
                        response = respuesta.text.lower()

                        print(Id,response)

                    except:

                        chat_completion = client.chat.completions.create(
                            messages=[
                                {
                                    "role": "user",
                                    "content": getQuery2(
                                              Text, Tag, Tag_Text),
                                }
                            ],
                            model="gpt-4o-mini",
                        )
                        response =
                           chat_completion.choices[0].message.content.lower()
                           print(Id,response,"gpt")

                # Escribe las etiquetas 'Id' y 'Reliability_Label'
                    en el archivo CSV
                if response == "confiable" or
                    response == "no confiable" or
                    response == "semiconfiable":
                    break
```

```
            csv_writer.writerow([Id, response])
```

3.3. Técnicas utilizadas de ingeniería de prompt

Como mencioné anteriormente, se emplearán las tres técnicas más utilizadas descritas en la guía oficial de ingeniería de prompt. A continuación, explicaré cómo se desarrolló el prompt para abordar el problema específico aplicando cada una de las técnicas y presentando el mejor ejemplo de los evaluados que ha obtenido el resultado más óptimo. En las tres técnicas, se ha seguido la sintaxis recomendada por la guía de prompts, utilizando separadores como almohadillas (###) para distinguir las instrucciones del contexto y asteriscos (**) para enfatizar partes del texto. También, el texto se ha dividido en varias partes detalladas para obtener el mejor resultado.

3.3.1. Prompt sin entrenamiento previo (Zero-shot)

Antes de presentar los resultados, es importante recordar que esta técnica se basa en generar respuestas sin necesidad de entrenar el modelo previamente con datos.

```
Python
PROMPT_TEMPLATE = """

    ### Instrucción ###

    Tu tarea es evaluar la fiabilidad de un fragmento de texto (`Tag_Text`)
en función de su categoría 5W1H (`5W1H_Label`).
    Debes clasificar la fiabilidad en una de las siguientes tres categorías:
"confiable", "semiconfiable" o "no confiable".

    ## Detalles de las Etiquetas ##

    - **Text**: La noticia completa de la cual el fragmento es parte.
    - **Tag_Text**: Un fragmento de la noticia que debes evaluar.
    - **5W1H_Label**: La categoría de la etiqueta que describe el tipo de
información contenida en el fragmento:
```

```
      - **WHAT (Qué)**: Información sobre hechos, objetos o detalles
específicos.
      - **WHO (Quién)**: Información sobre personas o entidades involucradas.
      - **WHEN (Cuándo)**: Información sobre fechas, tiempos o periodos.
      - **WHERE (Dónde)**: Información sobre lugares o ubicaciones.
      - **WHY (Por qué)**: Información sobre razones o causas.
      - **HOW (Cómo)**: Información sobre métodos o procedimientos.

    ## Criterios de Evaluación ##

    1. **Precisión**: Verifica si el contenido del fragmento es claro y puede
ser corroborado. La precisión se refiere a la exactitud de los detalles
proporcionados.

    2. **Objetividad**: Evalúa si el contenido es imparcial y libre de
sesgos. La objetividad implica que el texto presenta la información de manera
neutral y sin opiniones personales.

    ## Categorías de Fiabilidad ##

    - **confiable**: El fragmento es tanto preciso como objetivo.
    - **semiconfiable**: El fragmento cumple solo uno de los dos criterios
(precisión u objetividad).
    - **no confiable**: El fragmento no cumple con los criterios de precisión
ni de objetividad.

    ## Instrucciones Adicionales ##

    - Usa "semiconfiable" solo cuando estés seguro de que el fragmento cumple
solo uno de los criterios.
    - Responde con una sola palabra en minúsculas: "confiable",
"semiconfiable" o "no confiable". No incluyas puntuación adicional.

    """
```

Este prompt está estructurado en 5 secciones clave para guiar al modelo correctamente. En la primera sección, se establece claramente la tarea del modelo. Esta directriz asegura que el modelo debe evaluar la fiabilidad del fragmento del texto. Al definir la tarea de manera explícita, se asegura que el modelo entienda exactamente lo que se espera desde el inicio.

La segunda sección se enfoca en detallar las etiquetas que el modelo utilizará para llevar a cabo la evaluación. Aquí se describe cómo *Text* representa la noticia completa de la que el fragmento es parte, y cómo *Tag_Text* es el fragmento

específico que debe evaluarse. Además, se aclara que *5W1H_Label* categoriza el tipo de información contenida en el fragmento según las seis preguntas fundamentales del periodismo. Esta explicación ayuda al modelo a comprender el contexto y el tipo de información que debe analizar.

En la tercera sección, se describen los dos criterios clave que el modelo debe aplicar para la evaluación: precisión y objetividad. Estos criterios, obtenidos de la guía RUN-AS, son esenciales para determinar la fiabilidad del fragmento. La precisión se refiere a la claridad y verificabilidad del contenido, mientras que la objetividad evalúa la imparcialidad del texto. Estos criterios son fundamentales para que el modelo haga una clasificación adecuada.

La cuarta sección proporciona definiciones claras para las tres categorías de fiabilidad que el modelo debe utilizar en su clasificación.

Finalmente, en el último párrafo, se dan pautas específicas para asegurar respuestas uniformes del sistema. Se le indica que use "semiconfiable" solo cuando esté seguro de que solo se cumple uno de los criterios de evaluación y se especifica cómo debe generar sintácticamente la respuesta del modelo.

Podemos observar que para esta técnica, no se ha utilizado ninguna referencia externa. El sistema ha generado la respuesta basándose exclusivamente en el conocimiento con el que fue entrenado. Y, gracias a esta estructura detallada, el prompt ha logrado un resultado final de **0.51500** en el F1-score.

3.3.2. Prompt con pocas muestras (Few-shot)

A diferencia de la técnica anterior, esta se basa en ofrecer al modelo una pequeña cantidad de ejemplos para guiar sus respuestas en la tarea específica. La estructura utilizada a continuación, es similar a la utilizada en la técnica previa.

```python
PROMPT_TEMPLATE = """

    ### Instrucción ###
```

```
    Tu tarea es clasificar la fiabilidad de una noticia basada en la relación
entre un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). Las
categorías 5W1H son:

    - **WHAT (Qué)**: Hechos, objetos o elementos concretos mencionados.
    - **WHO (Quién)**: Personas o entidades mencionadas.
    - **WHEN (Cuándo)**: Tiempos o periodos mencionados.
    - **WHERE (Dónde)**: Lugares o direcciones específicas.
    - **WHY (Por qué)**: Razones o causas mencionadas.
    - **HOW (Cómo)**: Métodos o procedimientos detallados.

    Para cada noticia, considera lo siguiente:

    1. **Precisión**: ¿La información es específica y verificable? Ejemplos
de alta precisión incluyen fechas exactas o instituciones reconocidas.
       - Ejemplo: "El viernes 19 de marzo" es preciso (confiable).
       - Ejemplo: "Hace mucho tiempo" es impreciso (no confiable).

    2. **Objetividad**: ¿El texto es neutral y sin sesgo? Evalúa si el autor
expresa opiniones personales o si el texto tiene una carga emocional.
       - Ejemplo: "Según el rector de la Universidad de Alicante" es neutral
(confiable).
       - Ejemplo: "En mi opinión" es subjetivo (no confiable).

    Basado en estas consideraciones, clasifica la noticia en una de las
siguientes categorías de fiabilidad:

    - **confiable**: La información es precisa y objetiva.
    - **no confiable**: La información carece tanto de precisión como de
objetividad.
    - **semiconfiable**: La información cumple solo uno de los dos criterios
(precisión u objetividad).

    **Nota:** Evita usar "semiconfiable" a menos que estés seguro de que solo
se cumple uno de los criterios. Responde con una sola palabra en minúsculas:
"confiable", "semiconfiable" o "no confiable", sin puntos ni guiones.

    A continuación, te proporciono ejemplos específicos para ayudarte en la
clasificación:


    """
```

En primer lugar, el prompt define claramente la tarea: clasificar la fiabilidad de un fragmento de texto *Tag_Text* en función de su categoría 5W1H (*5W1H_Label)*.

Esta definición precisa ayuda al modelo a enfocar su análisis en los aspectos relevantes del texto desde el inicio.

A continuación, se detallan las etiquetas que el modelo debe utilizar. Esta sección proporciona un marco claro sobre cómo interpretar la información del fragmento y cómo cada categoría 5W1H ayuda a contextualizar el contenido. Esto es esencial para que el modelo comprenda el tipo de información que está evaluando y cómo encaja en el análisis general.

En cuanto a los criterios de evaluación, este prompt hace énfasis en la precisión y objetividad del contenido. La precisión se centra en la claridad y verificabilidad del fragmento, mientras que la objetividad evalúa la imparcialidad y la ausencia de sesgos. Estos criterios están claramente definidos y diferenciados, lo que guía al modelo en la evaluación de manera más rigurosa y específica en comparación con otros enfoques.

La siguiente parte proporciona definiciones precisas para las categorías de fiabilidad: “confiable”, “semiconfiable” y “no confiable”. Estas definiciones claras permiten al modelo clasificar el fragmento de manera coherente, asegurando que cada categoría esté bien comprendida y aplicada correctamente.

Finalmente, se establecen pautas detalladas para asegurar la uniformidad en las respuestas del modelo. Se especifica que "semiconfiable" debe usarse sólo cuando se cumpla uno de los criterios y se da una directriz sobre la forma en que debe ser la respuesta: una sola palabra en minúsculas sin puntuación adicional. Esta atención al detalle en la presentación de las respuestas ayuda a mantener la consistencia y claridad en los resultados.

La combinación de la tarea claramente definida, un marco de evaluación detallado, criterios específicos y pautas uniformes contribuye a que este prompt obtenga un F1-score de **0.52730**, reflejando su efectividad en la clasificación de la fiabilidad del texto.

3.3.3. Prompt cadena de pensamiento (CoT)

Para esta técnica se han generado dos prompts diferentes. Una empleando cadena de pensamiento sin ejemplos, “CoT-zero-shot”, y otra utilizando ejemplos, “CoT-few-shot”.

Empezaremos con la técnica que no requiere ejemplos. Recordemos que, esta técnica, está diseñada para que el modelo siga un conjunto específico de instrucciones detalladas que guían al proceso de evaluación.

```Python
PROMPT_TEMPLATE = """
 ### Instrucción

 Debes evaluar la fiabilidad de un subtexto (`Tag_Text`) en función de su correspondencia con una categoría 5W1H (`5W1H_Label`). Sigue estos pasos para hacer la evaluación:

 1. **Identificación del Subtexto y Categoría:**
    - Lee el `Tag_Text` y la `5W1H_Label` asignada.
    - Verifica que el subtexto se relacione específicamente con la categoría 5W1H asignada.

 2. **Evaluación de Precisión:**
    - **Confiable:** El subtexto proporciona información clara, detallada y verificable, con datos específicos.
    - **Semiconfiable:** El subtexto presenta algunos detalles claros, pero también contiene vaguedades.
    - **No confiable:** El subtexto es impreciso, vago o carece de detalles verificables.

 3. **Evaluación de Objetividad:**
    - **Confiable:** El subtexto es neutral, sin sesgos ni juicios de valor.
    - **Semiconfiable:** El subtexto muestra un leve sesgo o interpretación parcial.
    - **No confiable:** El subtexto es subjetivo, emocional o incluye juicios de valor.

 4. **Determinación de la Clasificación Final:**
    - **Confiable:** Si el subtexto cumple con ambos criterios (precisión y objetividad).
    - **Semiconfiable:** Si el subtexto cumple con solo uno de los criterios (precisión o objetividad).
    - **No confiable:** Si el subtexto no cumple con ninguno de los criterios.
```

```
    **Nota:** Responde con "confiable", "semiconfiable" o "no confiable".
Evita clasificar como "semiconfiable" a menos que esté claro que solo uno de
los criterios se cumple.

    """
```

El proceso comienza con una instrucción general que establece el propósito del modelo. A continuación, el modelo debe identificar y relacionar el subtexto con la categoría asignada, asegurándose de que exista una correspondencia clara y específica entre ambos.

La evaluación de la fiabilidad se realiza en dos dimensiones: precisión y objetividad. En cuanto a la precisión, el modelo clasifica el subtexto como “confiable”, “semiconfiable” o “no confiable”, dependiendo de si la información es clara, detallada y verificable, si contiene algunas vaguedades o si es imprecisa y carece de detalles verificables. La evaluación de la objetividad sigue un esquema similar, donde el subtexto es evaluado por su neutralidad y ausencia de sesgos o, por el contrario, por la presencia de subjetividad o juicios de valor.

Finalmente, el modelo determina la fiabilidad del subtexto considerando los criterios de precisión y objetividad. Según el grado en que se satisfagan estos criterios, la clasificación será “confiable”, “semiconfiable” o “no confiable”.

La implementación de este proceso ha dado como resultado un F1-score de **0.48253**, lo que indica la capacidad del sistema para clasificar la fiabilidad del subtexto con una precisión moderada.

La segunda técnica, “CoT-few-shot”, consiste en guiar al modelo de lenguaje para que evalúe la fiabilidad de un texto utilizando un enfoque estructurado con ejemplos previos (few-shot). Esto permite al modelo aprender y aplicar criterios de evaluación de manera más efectiva.

```Python
PROMPT_TEMPLATE = """
    ## Tarea ##
```

```
    Analiza el texto proporcionado y determina su fiabilidad. Asigna una
etiqueta a cada porción del texto según su **precisión** y **objetividad**.
Usa únicamente las etiquetas **"confiable,"** **"semiconfiable,"** o **"no
confiable."** Presenta el resultado sin puntuación adicional ni comentarios
adicionales.

    ## Proceso de Evaluación ##

    1. **Analiza el Texto:**
       - Identifica el subtexto relacionado con la etiqueta **5W1H_Label** y
evalúa su relevancia para la categoría correspondiente:
         - **WHAT (Qué):** Hechos, eventos, objetos o detalles específicos.
         - **WHO (Quién):** Personas, entidades u organizaciones mencionadas.
         - **WHEN (Cuándo):** Fechas, periodos o momentos específicos.
         - **WHERE (Dónde):** Lugares, ubicaciones o direcciones específicas.
         - **WHY (Por qué):** Razones, causas o motivaciones.
         - **HOW (Cómo):** Métodos, procedimientos o procesos.

    2. **Verifica la Precisión y la Objetividad:**
       - **Fuentes:** Examina si el subtexto cita fuentes reconocidas o
proporciona datos verificables.
       - **Exactitud:** Considera la especificidad y la claridad del
contenido presentado en el subtexto.
       - **Neutralidad:** Evalúa si el lenguaje es objetivo y libre de sesgos
o juicios emocionales.

    3. **Determina la Fiabilidad:**
       - **Confiable:** La información es precisa y objetiva, cumpliendo con
ambos criterios.
       - **Semiconfiable:** La información cumple con uno de los criterios
(precisión o objetividad), pero no con ambos.
       - **No confiable:** La información no cumple con ninguno de los
criterios (precisión ni objetividad).

    4. **Indicadores Adicionales:**
       - **Sesgos y Parcialidades:** Identifica cualquier posible sesgo o
parcialidad en el subtexto.
       - **Consistencia Interna:** Verifica si el subtexto es coherente
internamente.
       - **Contexto Relevante:** Considera cualquier contexto histórico o
cultural que pueda influir en la interpretación del subtexto.

    Aplica este enfoque sistemático para evaluar la fiabilidad del subtexto
proporcionado.

    ## Ejemplos de Evaluación ##
```

```
"""
```

Primero, el modelo debe identificar y analizar el subtexto relacionado con una etiqueta específica dentro del marco 5W1H. Este análisis se enfoca en categorías como hechos, personas, fechas, lugares, razones y métodos. La relación de cada porción del texto con estas categorías es esencial para determinar su relevancia.

Una vez que se ha identificado la categoría adecuada, el siguiente paso es verificar tanto la precisión como la objetividad del subtexto. Aquí, el modelo examina si el subtexto cita fuentes reconocidas o proporciona datos verificables, asegurando que el contenido sea específico, claro y libre de sesgos o juicios emocionales. Estos aspectos son fundamentales para evaluar la exactitud y neutralidad del texto.

Tras este análisis, el modelo determina la fiabilidad de la información. Si la información cumple con ambos criterios de precisión y objetividad, se clasifica como “confiable”. Si solo cumple con uno de los criterios, se clasifica como “semiconfiable”. Y, si no cumple con ninguno de los criterios, la clasificación resultante es “no confiable”.

Adicionalmente, se consideran otros indicadores relevantes, como la detección de posibles sesgos o parcialidades, la consistencia interna del texto y cualquier contexto histórico o cultural que pueda influir en la interpretación del subtexto. Estos factores adicionales son importantes para asegurar una evaluación completa y precisa.

Finalmente, se aplica un enfoque sistemático para evaluar la fiabilidad del texto proporcionado, incorporando ejemplos de evaluación que ayudan al modelo a su aprendizaje. Como resultado de la aplicación de esta técnica, se alcanzó un valor de F1-score de **0.49540**.

3.4. Hiperparámetros de configuración

Es posible ajustar el comportamiento y el rendimiento del modelo utilizando hiperparámetros específicos, lo que nos permite contrarrestar estos sesgos y

mejorar la precisión general del sistema. Entre los hiperparámetros más relevantes para los modelos utilizados en este proyecto, Gemini y ChatGPT, se encuentran el tamaño del lote, la tasa de aprendizaje, la temperatura y la probabilidad acumulativa entre otros. Estos parámetros desempeñan un papel crucial en cómo el modelo procesa la información y genera respuestas.

El tamaño del lote (*batch_size)* define cuántos ejemplos de datos utilizamos en cada iteración de entrenamiento. Ajustando este hiperparámetro, podemos influir en la estabilidad del entrenamiento y en la capacidad del modelo para generalizar. Los valores típicos varían entre pequeños (1 a 32), medios (64 a 256) y grandes (512 a 2048 o más). Un tamaño de lote grande permite un entrenamiento más estable y utiliza los recursos de manera más eficiente, aunque requiere más memoria y capacidad computacional. En cambio, un tamaño pequeño puede acelerar el aprendizaje y es útil cuando se dispone de menos memoria, aunque puede resultar en mayor variabilidad en la precisión.

La tasa de aprendizaje (*learning rate*) es otro hiperparámetro fundamental. Controla la velocidad a la que el modelo ajusta sus parámetros internos durante el entrenamiento. Los valores varían entre pequeños (1e-6 a 1e-4), medios (1e-4 a 1e-2) y grandes (1e-2 a 1e-1). Una tasa de aprendizaje alta puede acelerar la convergencia del modelo, pero también podría llevar a un sobreajuste o a la incapacidad de alcanzar un óptimo global. Por otro lado, una tasa de aprendizaje baja facilita ajustes más precisos en los parámetros aunque puede extender considerablemente el tiempo de entrenamiento.

La temperatura (*temperature*) es un hiperparámetro que afecta la creatividad o la aleatoriedad de las respuestas generadas por el modelo. Los valores oscilan entre 0.1 a 0.5 para respuestas más conservadoras y predecibles, ideales para aplicaciones que requieren consistencia y precisión. Valores medios, entre 0.6 y 1, ofrecen un equilibrio entre creatividad y coherencia, adecuados para la mayoría de las aplicaciones. Finalmente, temperaturas altas, de 1.1 a 2 o más, permiten generar respuestas más variadas y creativas, útiles en contextos donde se busca diversidad en las respuestas, aunque esto puede reducir la coherencia.

La probabilidad acumulativa (*top_p)*, también conocido como *nucleus sampling*, regula la diversidad de las respuestas limitando la probabilidad acumulativa de los tokens que el modelo considera. Con *top_p*, el modelo selecciona tokens de tal manera que la suma de las probabilidades de las palabras consideradas sea al menos igual a un umbral *p* especificado. Los valores varían entre 0 y 1: valores bajos, como 0.3, promueven respuestas más coherentes pero menos diversas, mientras que valores altos, como 1.0, permiten una mayor variedad y creatividad en las respuestas.

Después de realizar varias pruebas y ajustes, determinamos que la configuración predeterminada proporcionaba el mejor equilibrio entre precisión y rendimiento para nuestro modelo. Por lo tanto, optamos por mantener los hiperparámetros por defecto, ya que los resultados indicaron que esta configuración era la más efectiva y decidimos no realizar ajustes adicionales. En la configuración predeterminada, *batch_size* se establece en 1 durante la inferencia, *learning rate* para el entrenamiento es de 0.001, *temperature* se fija en 1 para ofrecer un equilibrio entre creatividad y coherencia en la generación de texto, y *top_p* se ajusta a 1 también, lo que permite al modelo considerar todas las opciones posibles al generar texto, promoviendo así una mayor diversidad en las respuestas [32].

3.5. Modelos generativos utilizados

En este proyecto se utilizarán modelos generativos avanzados, como Gemini y ChatGPT, basados en IA y diseñados para la generación y análisis automatizado de texto. Estos modelos, entrenados en grandes volúmenes de datos, permiten no solo la creación de contenido coherente, sino también la evaluación de la fiabilidad informativa, lo que los posiciona como herramientas fundamentales para la detección de desinformación y el análisis lingüístico de textos en español.

3.5.1. Gemini

Gemini es el modelo de IA más reciente y avanzado de Google Deepmind. Ha sido diseñado con capacidades multimodales desde su base y puede realizar un

razonamiento intermodal notable en textos, imágenes, videos, audio y código[33]. Está disponible en tres versiones:

- **Ultra**: Es el modelo más potente y es ideal para tareas altamente complejas.
- **Pro**: Es considerado el mejor modelo para una amplia gama de tareas. Este será el modelo utilizado para este proyecto.
- **Nano**: Es un modelo eficiente para tareas con limitaciones de memoria en el dispositivo, disponible en versiones de 1.8B (Nano-1) y 3.25B (Nano-2) parámetros, extraídos de modelos grandes de Gemini y cuantificados a 4 bits.

Gemini destaca en 20 pruebas de referencia multimodales y es el primer modelo en lograr un rendimiento humano experto en MMLU, una prueba que evalúa conocimientos en 57 materias, desde ciencia hasta humanidades, y desafía a los modelos a resolver problemas como lo haría una persona. Gemini Ultra obtiene un 90.0% en MMLU y un 62.4% en la prueba de referencia MMMU, que requiere conocimiento y razonamiento a nivel universitario.

Gemini Pro es ideal para aplicaciones donde se necesita un alto rendimiento sin la complejidad extrema que manejaría Gemini Ultra. A diferencia de la versión Nano, que está optimizada para dispositivos con limitaciones de memoria, Gemini Pro ofrece un rendimiento robusto sin los compromisos que vienen con modelos más ligeros. Por esta razón, la versión Pro ha sido seleccionada para el proyecto asegurando un rendimiento sólido y confiable.

Los modelos Gemini están diseñados para procesar información extensa de hasta 32.000 unidades de contexto, usando tecnología avanzada de decodificadores *Transformer* con mecanismos de atención eficiente. Esto les permite manejar texto, audio y elementos visuales simultáneamente, generando tanto texto como imágenes. Además, tienen una precisión del 98% en la recuperación de datos a lo largo del contexto completo, lo que es esencial para tareas como la recuperación de documentos y la comprensión de videos.

Aunque para este proyecto no se ha utilizado el modelo Gemini Ultra, hay que destacar que alcanza la máxima precisión cuando se combina con otros enfoques como la cadena de pensamiento (CoT) y la autoconsistencia, ya que ayudan a manejar la incertidumbre del modelo.

3.5.2. ChatGPT

ChatGPT es un avanzado modelo de lenguaje que permite interacciones más naturales y conversacionales con una computadora. GPT, acrónimo de “Generative Pre-trained Transformer” (Transformador Generativo Pre-entrenado), describe una familia de modelos de lenguaje natural desarrollados por OpenAI, una organización de investigación en IA.

Este modelo está entrenado con 570 GB de datos (300,000 millones de palabras) y 175,000 millones de parámetros incorporados. Su interfaz es fácil de usar, permitiendo a los usuarios solicitar datos, análisis e incluso opiniones. Al igual que Gemini, la tecnología de ChatGPT está basada en la arquitectura de "transformadores", capaz de generar texto original y preciso a partir de las preguntas de los usuarios.

La API de OpenAI ofrece una variedad de modelos con diferentes capacidades y rangos de precios. En un principio, para este proyecto se utilizó el modelo GPT-3.5-turbo-0125. Sin embargo, en julio de este año, se empleó un nuevo modelo más económico, más capaz, multimodal e igual de rápido llamado GPT-4o-mini.

GPT-4o-Mini no está diseñado para competir con la versión estándar, sino con modelos más pequeños como GPT-3.5 Turbo, que están diseñados para sistemas con menos recursos. Ambos tienen una ventana de contexto de 128,000 unidades pero la versión Mini es más ligera lo que la hace adecuada para tareas menos complejas ejecutándolas con mayor eficiencia.

Tener menos capacidad de procesamiento tiene la ventaja de que requiere menos recursos, permitiendo su uso en dispositivos con hardware limitado como móviles. Aunque no puede analizar textos tan profundamente ni crear artículos con el mismo nivel de coherencia que la versión estándar, su rapidez lo hace adecuado para aplicaciones móviles y chatbots básicos que manejan preguntas simples. La versión del modelo utilizada es más ligera y eficaz para clasificar textos cortos, ya que, aunque tiene una capacidad inferior, es suficiente para realizar el objetivo de este proyecto.

Se probó la versión premium mediante una suscripción de pago por tokens. Utilicé este modelo para realizar predicciones sobre todos los datos, pero no resultó ser lo suficientemente potente para este propósito ya que, consumía rápidamente los tokens incluidos en la suscripción mensual. Por lo tanto, se optó por utilizar este modelo como respaldo en caso de que Gemini generara alguna excepción durante el proceso de predicción.

A continuación, se presenta un gráfico circular con el modelo que mayormente ha utilizado el ChatGPT. Para que el gráfico sea más claro, se muestran únicamente dos valores: las etiquetas predichas por el modelo Gemini y las etiquetas predichas por ChatGPT eliminando otras etiquetas de fiabilidad. Al examinar el gráfico, es evidente que el uso de ChatGPT es considerablemente menor en comparación con Gemini. De hecho, sólo el 0,5% de las etiquetas predichas en el conjunto de datos de test provienen de ChatGPT, lo que equivale a aproximadamente 6 etiquetas de las 1.211 en total para el conjunto de datos de test (*5w1h_subtask_2_test.json).*

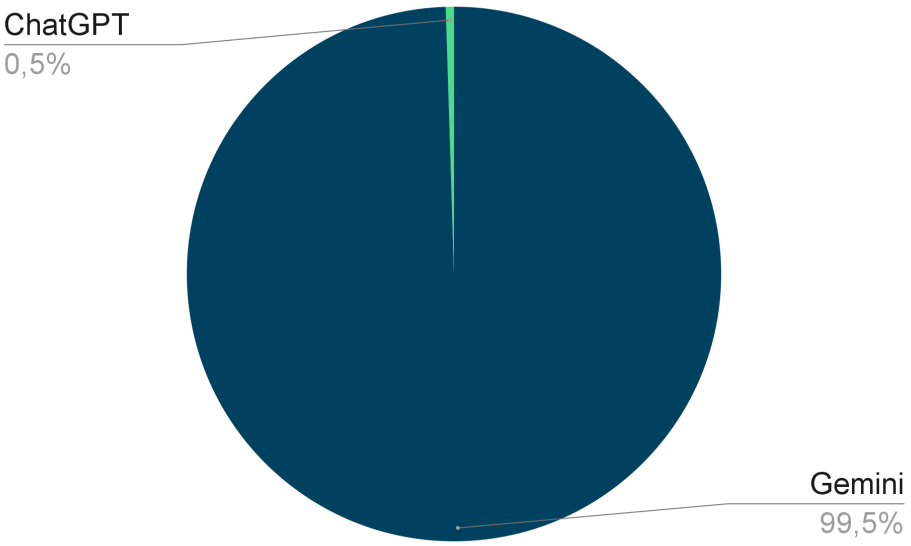

Ilustración 3.4 Distribución de etiquetas utilizadas con Gemini y con ChatGPT

3.6. Post-procesamiento

El post-procesamiento en este proyecto ha sido un paso crucial para asegurar la precisión y consistencia de los resultados. Después de ejecutar el código, evaluamos los resultados generados por el modelo, comparando las métricas obtenidas con los valores que esperábamos. Aunque el sistema estaba diseñado para clasificar los datos en tres categorías específicas ("confiable", "semiconfiable" y "no confiable"), en algunas ocasiones, produjo etiquetas distintas a las previstas.

Para solucionar esto, implementamos un script en Python que automatizó el proceso de corrección. El proceso es bastante sencillo: primero, cargamos el archivo CSV en un DataFrame utilizando la biblioteca *pandas*. Luego, seleccionamos la columna donde están las etiquetas de fiabilidad y obtenemos todos los valores únicos presentes en esa columna, además de contar cuántas etiquetas diferentes existen.

Después, reemplazamos las etiquetas incorrectas con su correspondiente correcta. Para este ejemplo, si había una etiqueta escrita como "confiables", el script la cambia a "confiable". Una vez hecho esto, guardamos el archivo corregido en un nuevo CSV para proceder a su evaluación.

```python
import pandas as pd

# Cargamos el archivo CSV en un DataFrame
df = pd.read_csv('C:/Users/elena/Desktop/2_few_shot/trial/archivo.csv')

# Seleccionamos columna que nos interesa
columna = df['Reliability_Label']

# Obtenemos los valores únicos de la columna y los contamos
etiquetas = columna.unique()
contador_etiquetas = len(etiquetas)

print(f'Hay {contador_etiquetas} valores diferentes en la columna
"{columna.name}" y son: {etiquetas}')

# Reemplazamos los valores incorrectos
df['Reliability_Label'] = df['Reliability_Label'].replace({
    'confiables': 'confiable',
    # 'valor_incorrecto2': 'semiconfiable',
```

```
    # 'valor_incorrecto3': 'no confiable'
})

# Volvemos a obtener los valores únicos después de reemplazarlos
etiquetas_actualizadas = df['Reliability_Label'].unique()
contador_etiquetas_actualizadas = len(etiquetas_actualizadas)
print(f'Se han reemplazado los valores incorrectos y las etiquetas actuales
son {contador_etiquetas_actualizadas}: {etiquetas_actualizadas}')

# Guardamos el archivo corregido
df.to_csv('C:/Users/elena/Desktop/2_few_shot/trial/archivo_actualizado.csv',
index=False)
```

3.7. Tecnologías utilizadas

En el desarrollo de este proyecto, se han utilizado varias tecnologías clave que han sido fundamentales para alcanzar los objetivos propuestos. Entre estas, destacan las siguientes:

- **Python:** se ha escogido como el lenguaje principal de programación debido a su versatilidad y su amplia gama de bibliotecas especializadas en PLN y machine learning. *Python* es un lenguaje de programación de código abierto y es conocido por su sintaxis clara y legible, lo que facilita el aprendizaje y la escritura de código, incluso para principiantes. Además de su uso en aplicaciones como desarrollo web, análisis de datos, IA y automatización cuenta con una amplia colección de bibliotecas y frameworks que lo convierte en la opción preferida para desarrolladores y científicos de datos. La comunidad de *Python* es extensa y activa, lo que asegura un continuo desarrollo, soporte y recursos educativos disponibles para usuarios de todos los niveles.

- **Pandas:** en el post-procesamiento de los datos, utilizamos esta librería de Python diseñada para manejar archivos en formatos como CSV y Excel. Esta herramienta facilita la lectura, manipulación y análisis de los datos, permitiéndonos corregir y ajustar las etiquetas de manera eficiente. Gracias a sus diversas funcionalidades, pudimos cargar los

datos, hacer los cambios necesarios y preparar la información asegurando así que los resultados fueran precisos y coherentes.

- **Jupyter Notebook:** esta herramienta ha sido esencial para el desarrollo iterativo y la experimentación rápida. Esta plataforma se descargó e instaló fácilmente a través de *Anaconda*, una distribución popular de *Python* que incluye una gran cantidad de paquetes y herramientas para la ciencia de datos. *Jupyter Notebook* nos ha permitido escribir el código y llevar a cabo la implementación completa del sistema de detección de fiabilidad de noticias en un entorno interactivo.

- **Visual Studio Code:** este entorno de desarrollo integrado (IDE) ha sido esencial para la evaluación del sistema, permitiéndonos calcular y analizar métricas de rendimiento como el F1-score, precisión, exactitud y recall que nos ha ayudado a mejorar el desarrollo del prompt. Esta plataforma destaca por su flexibilidad para trabajar con múltiples lenguajes de programación y su integración con herramientas de depuración y control de versiones. Su capacidad para manejar proyectos de gran escala y facilitar la colaboración entre desarrolladores contribuye a un análisis detallado y a una gestión eficiente del código, optimizando así el flujo de trabajo en el desarrollo de software.

- **Google Colabolatory:** es una herramienta en línea desarrollada por Google Research que permite escribir y ejecutar código *Python* directamente en el navegador. Está diseñada específicamente para el aprendizaje automático y el análisis de datos, utilizando un formato de cuaderno similar al de *Jupyter* para la ejecución de código en celdas. Lo que la distingue es su acceso limitado a recursos adicionales, como memoria RAM y GPUs, lo cual resulta especialmente útil para el entrenamiento de modelos, aprovechando la arquitectura paralela de las últimas GPUs para optimizar el proceso de entrenamiento. Inicialmente, empezamos utilizando este entorno pero debido a las limitaciones en el acceso a GPUs utilizamos *Jupyter Notebook*. A

diferencia de Colab, que ofrece GPUs en la nube, nuestro ordenador cuenta con GPUs más potentes. Esto nos permitió ejecutar el código con mayor eficacia.

Ilustración 3.5 Logos de las tecnologías utilizadas

4. RESULTADOS Y DISCUSIÓN

Los resultados del proyecto presentan una amplia variabilidad en la detección de fiabilidad de noticias falsas mediante la aplicación de modelos generativos de IA como Gemini y ChatGPT.

A continuación, haremos un análisis exhaustivo de los resultados obtenidos durante el desarrollo del sistema para cada una de las técnicas de prompting. Además, realizaremos un análisis de los errores en los que el modelo generativo no ha clasificado correctamente la fiabilidad de la noticia. Por último, llevaremos a cabo una comparación de los resultados obtenidos y los resultados de los demás participantes en la tarea FLARES (Clasificación de confiabilidad).

4.1. Análisis de resultados

A continuación, se presentan los resultados obtenidos con cada una de las técnicas de prompt utilizadas, evaluadas a través del valor *F1-score*.

Para probar cada una de estas técnicas, nos basaremos en el diagrama de actividad presentado en el apartado 3.2, donde se detallan varios pasos del proceso, centrándonos en la evaluación del sistema que corresponde al séptimo paso.

De acuerdo con el diagrama, se llevarán a cabo 10 pruebas con cada una de las cuatro técnicas de prompting (“zero-shot”, “few-shot”, “CoT-zero-shot” y “CoT-few-shot”). Esto hace un total de 40 pruebas o modelos desarrollados empleando el archivo *trial* (*5w1h_subtask_2_trial.json*) proporcionado en la competición FLARES. En el apartado 6.1, se muestra una tabla en la que se especifica cada prompt desarrollado y el modelo utilizado en cada técnica de prompt.

Los resultados de estas pruebas, nos permitirán identificar los tres prompts con el mejor desempeño. Estos tres mejores prompts, se utilizarán posteriormente para hacer las predicciones de fiabilidad de las noticias empleando el archivo de test (*5w1h_subtask_2_test.json*) con el objetivo de evaluar el desempeño final del modelo en nuestro sistema de clasificación de noticias.

A continuación, se llevará a cabo un análisis de todos los modelos desarrollados para comparar y verificar su eficacia en el sistema. Estos análisis se presentarán en tablas comparativas, ordenadas de mayor a menor rendimiento, con el fin de facilitar la visualización y comparación de los resultados.

4.1.1. Resultados obtenidos con el archivo de prueba

Se presentan las tablas con los resultados obtenidos utilizando el archivo de prueba para cada técnica de prompt utilizada. A través del análisis de esta tabla, se busca identificar los tres prompts más efectivos al comparar las métricas obtenidas en cada modelo.

Las siguientes tablas presentan una estructura de cuatro columnas y diez filas. La primera columna, titulada "Modelos", está destinada al nombre del modelo

evaluado, por ejemplo, "modelo x" (se ha elegido este nombre para facilitar la interpretación de la tabla). Las tres columnas restantes muestran las métricas de evaluación del modelo: exactitud (*accuracy*), precisión (*precision*) y cobertura (*recall*). Además, se incluye el *F1-score*, que será la métrica principal para evaluar el desempeño del modelo. Para los resultados, resaltaremos en verde los tres mejores y en rojo el peor de todos ellos.

4.1.1.1. Resultados obtenidos con la técnica “zero-shot”

Modelos	Accuracy	Precision	Recall	F1-score
Modelo 1	0.66606	0.45255	0.48706	**0.45875**
Modelo 2	0.62613	0.44867	0.48133	**0.44667**
Modelo 3	0.64609	0.42834	0.46703	**0.43294**
Modelo 4	0.62250	0.42835	0.44922	**0.42840**
Modelo 5	0.61705	0.42781	0.46360	**0.42772**
Modelo 6	0.54627	0.43905	0.59195	**0.41902**
Modelo 7	0.60980	0.40924	0.45824	**0.41089**
Modelo 8	0.64609	0.40711	0.42763	**0.41001**
Modelo 9	0.57531	0.40397	0.46640	**0.39593**
Modelo 10	0.68602	0.36079	0.36894	**0.36287**

Tabla 4.1 Resultados obtenidos con la técnica “zero-shot”

En general, al evaluar los modelos en la tabla, se observa una variedad significativa en su desempeño en términos de *accuracy*, *precisión*, *recall* y *F1-score*.

En cuanto al ***accuracy***, los modelos con los valores más altos, como el *Modelo 10* (0.686) y el *Modelo 1* (0.666), muestran un buen rendimiento en la clasificación general. Sin embargo, estos modelos no siempre son los mejores en otras métricas, lo que indica que un alto *accuracy* no siempre se traduce en un buen equilibrio general en la clasificación.

La ***precisión*** de los modelos muestra una tendencia similar. El *Modelo 1* y el *Modelo 2* tienen las mejores precisiones, indicando que son más efectivos en identificar correctamente las instancias positivas. Por otro lado, el *Modelo 10*, a pesar de su alto accuracy, tiene la precisión más baja, sugiriendo que puede estar cometiendo más errores al clasificar las instancias positivas.

En términos de ***recall***, el *Modelo 6* destaca con el valor más alto, lo que indica una fuerte capacidad para detectar instancias positivas entre las verdaderas positivas. Los modelos con altos valores de recall, como el *Modelo 1* y el *Modelo 2*, también muestran una buena capacidad para identificar la mayoría de las instancias relevantes.

4.1.1.2. Resultados obtenidos con la técnica “few-shot”

Modelos	Accuracy	Precision	Recall	F1-score
Modelo 1	0.65698	0.44682	0.51805	**0.46242**
Modelo 2	0.66424	0.43695	0.47747	**0.44418**
Modelo 3	0.60254	0.41523	0.48999	**0.42312**
Modelo 4	0.63157	0.41893	0.47021	**0.41824**
Modelo 5	0.65517	0.40272	0.44682	**0.41073**
Modelo 6	0.58076	0.41094	0.45709	**0.40925**
Modelo 7	0.5807	0.41094	0.45709	**0.40904**
Modelo 8	0.53176	0.43219	0.50687	**0.40729**
Modelo 9	0.65880	0.39777	0.44001	**0.40628**
Modelo 10	0.68239	0.38836	0.38687	**0.38007**

Tabla 4.2 Resultados obtenidos con la técnica “few-shot”

En general, al analizar la tabla de los modelos, se observan diferencias notables en términos de las cuatro métricas de evaluación, lo que sugiere que cada modelo tiene sus propias fortalezas y debilidades.

La exactitud o ***accuracy*** muestra que el *Modelo 10* tiene el valor más alto (0.6823), lo que indica un buen desempeño general en la clasificación. Sin embargo, este modelo tiene un bajo *F1-score*, lo que sugiere que su alto accuracy no se traduce en un equilibrio efectivo entre precisión y recall. En contraste, modelos como el *Modelo 1* y el *Modelo 2* también presentan un alto accuracy (0.6569 y 0.6642, respectivamente), y sus *F1-score* son más altos en comparación con el *Modelo 10*, lo que indica un mejor equilibrio general.

En términos de ***precisión***, los modelos con las mejores métricas son el *Modelo 1* y el *Modelo 2*, con valores de 0.4468 y 0.4369, respectivamente. Estos modelos no solo tienen una buena precisión, sino que también se desempeñan bien en otras métricas, lo que sugiere una mayor efectividad en la identificación correcta de instancias positivas. Por otro lado, el *Modelo 10* tiene la precisión más baja (0.388), lo que indica que podría estar cometiendo más errores al clasificar instancias positivas.

El ***recall*** revela que el *Modelo 1* tiene el valor más alto (0.518), mostrando una fuerte capacidad para identificar instancias positivas. El *Modelo 2* también tiene un buen recall (0.4777), lo que sugiere que ambos modelos son efectivos en detectar instancias relevantes. En contraste, modelos como el *Modelo 10* tienen un recall más bajo (0.3868), indicando que tienen dificultades para detectar la mayoría de las instancias positivas.

El ***F1-score*** proporciona una visión integral del rendimiento general de los modelos. El *Modelo 1* destaca con el *F1-score* más alto (0.4624), lo que sugiere un buen equilibrio entre precisión y recall. El *Modelo 2* también tiene un *F1-score* competitivo (0.4441), reflejando un equilibrio efectivo. En comparación, el *Modelo 10* tiene el *F1-score* más bajo (0.38), lo que indica un equilibrio deficiente entre precisión y recall.

4.1.1.3. Resultados obtenidos con la técnica “CoT-zero-shot”

Modelos	Accuracy	Precision	Recall	F1-score
Modelo 1	0.70054	0.43905	0.50079	**0.45574**
Modelo 2	0.72776	0.43098	0.42597	**0.41941**
Modelo 3	0.64972	0.40887	0.44125	**0,41834**
Modelo 4	0.61887	0.41308	0.47645	**0.41677**
Modelo 5	0.66424	0.41230	0.42924	**0.41151**
Modelo 6	0.62794	0.41546	0.46917	**0.41036**
Modelo 7	0.66243	0.40410	0.42570	**0.40827**
Modelo 8	0.62976	0.42819	0.40683	**0.40809**
Modelo 9	0.64428	0.40373	0.45382	**0.40324**
Modelo 10	0.74410	0.42281	0.39624	**0.40118**

Tabla 4.3 Resultados obtenidos con la técnica “CoT-zero-shot”

En esta tabla, la exactitud o ***accuracy*** muestra que el *Modelo 10* tiene el valor más alto (0.7441), indicando el mejor desempeño general en términos de clasificaciones correctas. Sin embargo, un alto *accuracy* no siempre se traduce en un buen equilibrio entre precisión y *recall*. Modelos como el *Modelo 2* y el *Modelo 1* también presentan altos valores de *accuracy* (0.727 y 0.7), con el *Modelo 1* mostrando un equilibrio más favorable en otras métricas.

En cuanto a ***precisión***, el *Modelo 1* lidera con un valor de 0.439, lo que sugiere una alta capacidad para clasificar correctamente las instancias positivas. Los modelos *Modelo 2* y el *Modelo 6* también tienen buenas precisiones (0.43 y 0.415, respectivamente). En contraste, el *Modelo 7* tiene la precisión más baja (0.404), indicando que podría estar clasificando incorrectamente una mayor proporción de instancias positivas.

El ***recall*** revela que el *Modelo 4* tiene el valor más alto (0.476), lo que indica una mejor capacidad para identificar instancias positivas entre las verdaderas positivas. Modelos como el *Modelo 6* y el *Modelo 1* también tienen valores de recall

competitivos (0.4691 y 0.5007), sugiriendo una buena capacidad para detectar instancias relevantes. En contraste, el *Modelo 10* tiene el recall más bajo (0.396), indicando una menor capacidad para identificar instancias positivas.

El valor ***F1-score*** ofrece una perspectiva integral del equilibrio entre precisión y recall. El *Modelo 1* tiene el *F1-score* más alto (0.455), indicando un buen equilibrio entre precisión y recall. Modelos como el *Modelo 2* y el *Modelo 5* también muestran buenos valores de *F1-score* (0.419 y 0.412, respectivamente). En comparación, el *Modelo 10* tiene el *F1-score* más bajo (0.401), lo que sugiere que el modelo no mantiene un equilibrio efectivo entre precisión y recall, a pesar de su alto accuracy.

4.1.1.4. Resultados obtenidos con la técnica "CoT-few-shot"

Modelos	Accuracy	Precision	Recall	F1-score
Modelo 1	0.67876	0.44716	0.56025	**0.46300**
Modelo 2	0.60798	0.43902	0.52613	**0.44987**
Modelo 3	0.60072	0.44653	0.57734	**0.44898**
Modelo 4	0.63702	0.42797	0.49524	**0.43479**
Modelo 5	0.60072	0.42143	0.48040	**0.42821**
Modelo 6	0.61887	0.42058	0.46763	**0.42671**
Modelo 7	0.61524	0.41306	0.45122	**0.41182**
Modelo 8	0.72413	0.41832	0.42215	**0.40816**
Modelo 9	0.72232	0.40842	0.47711	**0.39270**
Modelo 10	0.55353	0.40311	0.47764	**0.39077**

Tabla 4.4 Resultados obtenidos con la técnica "CoT-few-shot"

Podemos observar que la exactitud o ***accuracy*** muestra que el *Modelo 8* tiene el valor más alto (0.724), seguido de cerca por el *Modelo 9* (0.722). Estos modelos son los mejores en términos de clasificación general correcta. Sin embargo, un alto accuracy no siempre implica un buen equilibrio en otras métricas. En

contraste, el *Modelo 10* tiene el accuracy más bajo (0.554), indicando un menor rendimiento general en la clasificación correcta.

En términos de ***precisión***, el *Modelo 3* lidera con el valor más alto (0.447), lo que indica una buena capacidad para clasificar correctamente las instancias positivas. El *Modelo 2* y el *Modelo 1* también presentan buenas precisiones (0.439 y 0.447, respectivamente). Por otro lado, el Modelo 10 tiene la precisión más baja (0.403), sugiriendo que podría estar cometiendo más errores en la identificación de instancias positivas.

El ***recall*** revela que el *Modelo 3* tiene el valor más alto (0.577), lo que indica una fuerte capacidad para detectar instancias positivas. El *Modelo 1* también muestra un buen recall (0.560), mientras que el *Modelo 2* y el *Modelo 4* tienen valores competitivos (0.526 y 0.495). En contraste, el *Modelo 10* tiene un recall de 0.478, lo que indica una menor capacidad para identificar instancias positivas en comparación con otros modelos.

El ***F1-score*** proporciona una visión equilibrada del rendimiento general. El *Modelo 2* tiene el *F1-score* más alto (0.450), seguido de cerca por el *Modelo 3* (0.449) y el *Modelo 1* (0.463). Estos modelos muestran un buen equilibrio entre precisión y recall. En comparación, el *Modelo 10* tiene el *F1-score* más bajo (0.391), lo que sugiere un desequilibrio en precisión y recall y un rendimiento general menos óptimo.

4.1.2. Resultados obtenidos con el archivo de test

A continuación, se presentan las tablas con los resultados obtenidos utilizando el archivo de test para cada técnica de prompt. El análisis de estas tablas tiene como objetivo identificar qué prompt ha logrado el mejor desempeño en cada técnica comparando las métricas obtenidas.

Las siguientes tablas presentan una estructura de seis columnas y tres filas. En la tabla anterior ya se presentan todas estas columnas: la primera, titulada "Modelos", con el nombre del modelo evaluado (por ejemplo, "modelo x"), y las cuatro siguientes que muestran las métricas de exactitud (*accuracy*), precisión

(*precision*), cobertura (*recall*) y *F1-score* como métrica principal de evaluación. En esta nueva tabla se agrega una columna adicional titulada "Fase" que hace referencia a la etapa de evaluación en la que nos encontramos: si estamos en la fase de prueba o en la de test.

Esta nueva columna ("fase"), ayuda a visualizar la clasificación de los resultados obtenidos de las pruebas anteriores realizadas con el conjunto de prueba o "*5w1h_subtask_2_trial.json"* y de las obtenidas ahora con el conjunto de test o "*5w1h_subtask_2_test.json*". Para los resultados, resaltaremos en verde el que ha obtenido mejor resultado en el *F1-score* y en rojo el peor. Este resaltado se aplicará únicamente a los modelos en la fase de test.

Es importante señalar que, para realizar el análisis comparativo de las siguientes tablas, los organizadores de la tarea FLARES[31] me dieron acceso completo al conjunto de datos de test. Este dataset es privado y fue exclusivamente proporcionado para esta investigación.

4.1.2.1. Resultados obtenidos con la técnica "zero-shot"

Modelos	Fase	Accuracy	Precision	Recall	F1-score
Modelo 1	test	0.57638	0.53778	0.52558	**0.51500**
Modelo 2	test	0.56895	0.51777	0.49171	**0.48138**
Modelo 3	test	0.53592	0.49838	0.49490	**0.47765**
Modelo 1	trial	0.66606	0.45255	0.48706	0.45875
Modelo 2	trial	0.62613	0.44867	0.48133	0.44667
Modelo 3	trial	0.64609	0.42834	0.46703	0.43294

Tabla 4.5 Comparativa de resultados obtenidos con la técnica "zero-shot"

Como podemos observar en esta primera tabla, en el *Modelo 1* muestra un aumento en el *F1-score* al pasar de la fase de trial a la fase de test, de 0.45875 a 0.51500. Aunque el *accuracy* disminuye de 0.66606 en la fase de trial a 0.57638 en

la fase de test, el modelo ha mejorado su precisión (de 0.45255 a 0.53778) y *recall* (de 0.48706 a 0.52558). Esta mejora en el *F1-score* indica que el modelo ha logrado un mejor equilibrio entre precisión y recall en la fase de test, a pesar de la reducción en *accuracy*.

El *Modelo 2* experimenta un ligero aumento en el *F1-score*, de 0.44667 en la fase de trial a 0.48138 en la fase de test. A pesar de la disminución en el *accuracy*, que baja de 0.62613 a 0.56895, el modelo ha logrado mejorar su precisión (de 0.44867 a 0.51777) y mantener un *recall* razonable (de 0.48133 a 0.49171). Esta mejora en el *F1-score* sugiere que el modelo ha ajustado bien sus predicciones en la fase de test, mostrando un rendimiento ligeramente mejor en términos de equilibrio entre precisión y *recall*.

El *Modelo 3* muestra un aumento en el *F1-score,* pasando de 0.43294 en la fase de trial a 0.47765 en la fase de test. Aunque el *accuracy* disminuye de 0.64609 a 0.53592, el modelo ha logrado una ligera mejora en *recall* (de 0.46703 a 0.49490) y una pequeña mejora en precisión (de 0.42834 a 0.49838). A pesar de la mejora en el *F1-score*, el modelo sigue teniendo el menor rendimiento general en términos de precisión y recall comparado con los otros dos modelos, y su menor accuracy refleja un desafío persistente en la clasificación general.

A modo de conclusión se puede observar que todos los modelos muestran una mejora en el *F1-score* en la fase de test en comparación con la fase de trial, indicando una capacidad más equilibrada para clasificar instancias positivas en la fase de test. Esto sugiere que, a pesar de las variaciones en *accuracy*, todos los modelos han logrado ajustes positivos en su capacidad de clasificación en la fase de test.

4.1.2.2. Resultados obtenidos con la técnica “few-shot”

Modelos	Fase	Accuracy	Precision	Recall	F1-score
Modelo 1	test	0.61023	0.52667	0.54131	**0.52730**
Modelo 2	test	0.58216	0.54233	0.52860	**0.51715**
Modelo 3	test	0.54417	0.50024	0.49904	**0.48177**
Modelo 1	trial	0.65698	0.44682	0.51805	0.46242
Modelo 2	trial	0.66424	0.43695	0.47747	0.44418
Modelo 3	trial	0.60254	0.41523	0.48999	0.42312

Tabla 4.6 Comparativa de resultados obtenidos con la técnica “few-shot”

El *Modelo 1* muestra una mejora en el *F1-score* al pasar de la fase de trial a la fase de test, aumentando de 0.4624 en la fase de trial a 0.5273 en la fase de test. Aunque el *accuracy* disminuye ligeramente de 0.65698 a 0.61023, el modelo ha mejorado tanto en precisión (de 0.44682 a 0.52667) como en *recall* (de 0.51805 a 0.54131). Esta mejora en el *F1-score* sugiere que el modelo ha optimizado su capacidad para equilibrar precisión y recall en la fase de test, resultando en una mayor efectividad en la clasificación.

El Modelo 2 también muestra una mejora en el *F1-score*, aumentando de 0.44418 en la fase de trial a 0.5171 en la fase de test. A pesar de una caída en el *accuracy*, que pasa de 0.66424 a 0.58216, el modelo ha incrementado su precisión (de 0.43695 a 0.54233) y ha mantenido un *recall* razonable (de 0.47747 a 0.52860). Este aumento en el *F1-score* indica que el modelo ha ajustado eficazmente sus predicciones en la fase de test, logrando un mejor equilibrio entre precisión y *recall* a pesar de la reducción en *accuracy*.

El *Modelo 3* muestra un aumento en el *F1-score* de 0.42312 en la fase de trial a 0.48177 en la fase de test. Aunque el *accuracy* disminuye de 0.60254 a 0.54417, el modelo ha mejorado ligeramente en *recall* (de 0.48999 a 0.49904) y en precisión (de 0.41523 a 0.50024). A pesar de esta mejora en el *F1-score*, el *Modelo 3* sigue mostrando el menor rendimiento en términos de precisión y *recall* comparado con los

otros dos modelos, lo que sugiere que, aunque ha habido una mejora, todavía enfrenta desafíos en la capacidad de clasificación general.

En definitiva, todos los modelos han demostrado mejoras en el *F1-score* en la fase de test en comparación con la fase de trial, indicando una mejor capacidad para equilibrar precisión y *recall*. El *Modelo 1* ha mostrado la mayor mejora en el *F1-score*, destacándose por un buen equilibrio entre precisión y *recall*. El *Modelo 2* también ha tenido una mejora notable, mientras que el *Modelo 3* presenta una mejora en el *F1-score*, pero sigue enfrentando retos en la precisión y *recall*. Esto refleja que, a pesar de las variaciones en *accuracy*, los modelos han logrado ajustar su rendimiento en la fase de test, con el *Modelo 1* logrando el mejor equilibrio general.

4.1.2.3. Resultados obtenidos con la técnica “CoT-zero-shot”

Modelos	Fase	Accuracy	Precision	Recall	F1-score
Modelo 1	test	0,45574	0.61354	0.48160	**0.48394**
Modelo 2	test	0,41941	0.60033	0.48537	**0.47347**
Modelo 3	test	0,41834	0.58464	0.47910	**0.45493**
Modelo 1	trial	0.70054	0.43905	0.50079	0.45574
Modelo 2	trial	0.72776	0.43098	0.42597	0.41941
Modelo 3	trial	0.64972	0.40887	0.44125	0,41834

Tabla 4.7 Comparativa de resultados obtenidos con la técnica “CoT-few-shot”

Observamos para el *Modelo 1*, se observa un ligero aumento en el F1-score al pasar de la fase de trial a la fase de test, subiendo de 0.45574 a 0.48394. Aunque el *accuracy* disminuye significativamente de 0.70054 a 0.45574, el modelo ha mostrado una mejora en precisión (de 0.43905 a 0.61354) y un leve ajuste en *recall* (de 0.50079 a 0.48160). Este cambio en el *F1-score* indica que, a pesar de la reducción en *accuracy*, el modelo ha logrado un mejor equilibrio entre precisión y *recall* en la fase de test.

El *Modelo 2* también experimenta un aumento en el *F1-score*, que pasa de 0.41941 en la fase de trial a 0.47347 en la fase de test. Aunque el *accuracy* baja de 0.72776 a 0.41941, el modelo ha mejorado en precisión (de 0.43098 a 0.60033) y en recall (de 0.42597 a 0.48537). Este aumento en el *F1-score* sugiere una optimización en el balance entre precisión y *recall* durante la fase de test, a pesar de la notable caída en *accuracy*.

El *Modelo 3* muestra un incremento en el F1-score, ascendiendo de 0.41834 en la fase de trial a 0.45493 en la fase de test. A pesar de que el *accuracy* disminuye de 0.64972 a 0.41834, el modelo ha incrementado su precisión (de 0.40887 a 0.58464) y su *recall* (de 0.44125 a 0.47910). La mejora en el *F1-score* indica que el modelo ha ajustado positivamente su rendimiento en la fase de test, aunque sigue presentando desafíos en comparación con los otros modelos en términos de precisión y *recall*.

Podemos apreciar que todos los modelos han mostrado mejoras en el *F1-score* durante la fase de test, lo que indica que han logrado un mejor equilibrio entre precisión y *recall*. El *Modelo 2* destaca con la mayor mejora en el *F1-score*, mientras que el *Modelo 1* también ha mostrado un ajuste positivo, a pesar de la reducción en accuracy. El *Modelo 3* presenta una mejora en el *F1-score*, pero sigue enfrentando desafíos en comparación con los otros modelos.

4.1.2.4. Resultados obtenidos con la técnica “CoT-few-shot”

Modelos	Fase	Accuracy	Precision	Recall	F1-score
Modelo 1	test	0.53922	0.49865	0.54698	**0.49540**
Modelo 2	test	0.54170	0.49689	0.52657	**0.48896**
Modelo 3	test	0.55656	0.46959	0.51212	**0.47202**
Modelo 1	trial	0.67876	0.44716	0.56025	0.46300
Modelo 2	trial	0.60798	0.43902	0.52613	0.44987
Modelo 3	trial	0.60072	0.44653	0.57734	0.44898

Tabla 4.8 Comparativa de resultados obtenidos con la técnica “CoT-few-shot”

Para el *Modelo 1*, se observa un ligero aumento en el *F1-score* al pasar de la fase de trial a la fase de test, subiendo de 0.463 en la fase de trial a 0.4954 en la fase de test. A pesar de una disminución en *accuracy* (de 0.67876 a 0.53922), el modelo ha mejorado en precisión (de 0.44716 a 0.49865) y *recall* (de 0.56025 a 0.54698). Este aumento en el F1-score sugiere una optimización en el equilibrio entre precisión y *recall*, a pesar de la caída en *accuracy*.

El *Modelo 2* también muestra un aumento en el *F1-score*, que pasa de 0.44987 en la fase de trial a 0.48896 en la fase de test. Aunque el *accuracy* mejora ligeramente de 0.60798 a 0.54170, el modelo ha incrementado su precisión (de 0.43902 a 0.49689) y ajustado su *recall* (de 0.52613 a 0.52657). Esta mejora en el *F1-score* indica que el modelo ha logrado un mejor balance entre precisión y *recall* en la fase de test, a pesar de un ligero aumento en *accuracy*.

El *Modelo 3* presenta una mejora en el *F1-score*, que sube de 0.44898 en la fase de trial a 0.47202 en la fase de test. Aunque el *accuracy* también mejora de 0.60072 a 0.55656, el modelo ha reducido su precisión (de 0.44653 a 0.46959) mientras que el *recall* ha disminuido ligeramente (de 0.57734 a 0.51212). La mejora en el *F1-score* refleja un ajuste positivo en el equilibrio entre precisión y *recall* durante la fase de test, a pesar de las ligeras caídas en precisión y recall.

Se puede notar que todos los modelos han mostrado mejoras en el *F1-score* en la fase de test en comparación con la fase de trial. El *Modelo 1* ha logrado un aumento notable en el *F1-score*, a pesar de la reducción en *accuracy* al igual que el *Modelo 2* y un ajuste equilibrado en precisión y *recall*, mientras que el *Modelo 3* presenta una mejora en el *F1-score*, aunque con ligeras caídas en precisión y *recall*.

4.1.3. Conclusiones finales sobre la realización de las pruebas

En todas estas tablas comparativas, podemos observar que los valores de las métricas de evaluación son similares entre sí, ya que no varían mucho. Esto sugiere que, en general, los modelos tienen un rendimiento relativamente homogéneo en términos de precisión, *recall* y *F1-score*, independientemente de las fases de trial o test.

Primero, es notable que en todos los modelos, el *F1-score* en la fase de test muestra una mejora respecto a la fase de trial, indicando que los modelos han sido capaces de optimizar el equilibrio entre precisión y *recall* al enfrentar datos nuevos. Este patrón de mejora sugiere que los modelos, en su mayoría, han aprendido a ajustar sus predicciones de manera más efectiva, alcanzando un mejor rendimiento en la fase de test.

Es importante destacar que, en cada uno de los 10 modelos correspondientes a cada técnica, el prompt se redacta de manera distinta para evaluar cómo varía su comportamiento con diferentes instrucciones. Para ello, se realizaron ajustes en el rediseño del prompt a medida que se obtenían diversos resultados. El objetivo de estos ajustes fue mejorar el *F1-score*, refinando tanto la redacción como las instrucciones proporcionadas en función de los resultados obtenidos.

A continuación mostramos una tabla de los resultados del modelo con el mejor rendimiento obtenido para cada técnica de ingeniería de prompts. Se debe recordar que estos mejores prompts se describen y explican detalladamente en el apartado 3.3.

Técnicas	F1-score
zero-shot	0.51500
few-shot	**0.52730**
CoT-zero-shot	0.48394
CoT-few-shot	0.49540

Tabla 4.9 Resultados mejores de las técnicas

Observamos que la técnica *few-shot* ha logrado obtener finalmente el mejor valor de *F1-score*, con un rendimiento de **0.5273**, superando a las demás técnicas evaluadas. Este resultado destaca la efectividad del enfoque *few-shot* en la tarea de clasificación de fiabilidad en las noticias, ya que logra un equilibrio óptimo entre precisión y *recall*. Mientras tanto, la técnica *zero-shot* también muestra un rendimiento sólido, pero con un *F1-score* ligeramente inferior de 0.515. Las técnicas *CoT-zero-shot* y *CoT-few-shot* obtuvieron valores de *F1-score* menores, indicando que estas técnicas pueden no ser tan efectivas en el contexto evaluado.

Analizando todos los desarrollos de prompt realizados (mostrados en el apartado 6.1) podemos concluir que la técnica “few-shot” ha demostrado un mejor desempeño ya que, además de incluir ejemplos específicos antes de predecir la confiabilidad, ha aplicado las recomendaciones de sintaxis proporcionadas por la guía de prompting[14]. También influye la claridad de la instrucción y cómo se ha redactado. Como hemos explicado más arriba, no podemos prever con exactitud cómo responderá el modelo generativo. Si el prompt se redacta de manera diferente, los resultados obtenidos pueden variar significativamente.

4.2. Análisis de errores

El análisis de errores es esencial para evaluar y mejorar la precisión de nuestro sistema de clasificación de confiabilidad. Consiste en identificar y entender los casos en los que el modelo no ha clasificado correctamente la fiabilidad de las noticias. Para ello haremos uso de las matrices de confusión, empleadas

exhaustivamente en el proyecto para detectar patrones y explorar las causas de estos errores y así, optimizar el rendimiento del sistema.

Es difícil determinar cómo se va a comportar el modelo cuando le indicamos una instrucción. Esto dependerá de varios factores, incluyendo la calidad y diversidad de los datos de entrenamiento, la precisión de la instrucción proporcionada, y la capacidad del modelo para interpretar y generalizar a partir de los ejemplos que ha aprendido. En particular los modelos generativos utilizados, *Gemini* y *ChatGPT,* cuentan con una base de datos amplia y diversa, lo que les permite tener un entendimiento enriquecido y matizado de diferentes contextos y temas.

Sin embargo, la eficacia de los modelos también dependerá de cómo se estructuren las instrucciones y de su capacidad para manejar ambigüedades o variaciones en los datos. Como hemos observado en el apartado 4.1, la interacción entre estos factores puede afectar significativamente el desempeño del modelo haciendo que los resultados sean impredecibles y requieran ajustes finos para lograr un comportamiento óptimo.

Los dos siguientes apartados 4.2.1 y 4.2.2, analizaremos la matriz de confusión y detectaremos patrones erróneos en la clasificación del **modelo que ha obtenido el mejor rendimiento** de todos los 40 modelos desarrollados utilizando el archivo de prueba o “trial”. Este modelo obtuvo un resultado final de **0.4630** en la métrica *F1-score* en la técnica “CoT-few-shot”.

4.2.1. Matriz de confusión

La matriz de confusión es una herramienta que permite evaluar el rendimiento de un sistema de clasificación al mostrar cómo las predicciones del modelo se comparan con las etiquetas reales. Esta matriz organiza las predicciones en una tabla que muestra la cantidad de casos correctamente clasificados y los errores cometidos por el modelo, lo que facilita identificar qué clases están siendo confundidas entre sí.

En el proyecto, esta matriz se ha utilizado para mejorar el desarrollo del prompt y su resultado para el archivo de prueba. Como hemos comentado a continuación la matriz de confusión del mejor rendimiento:

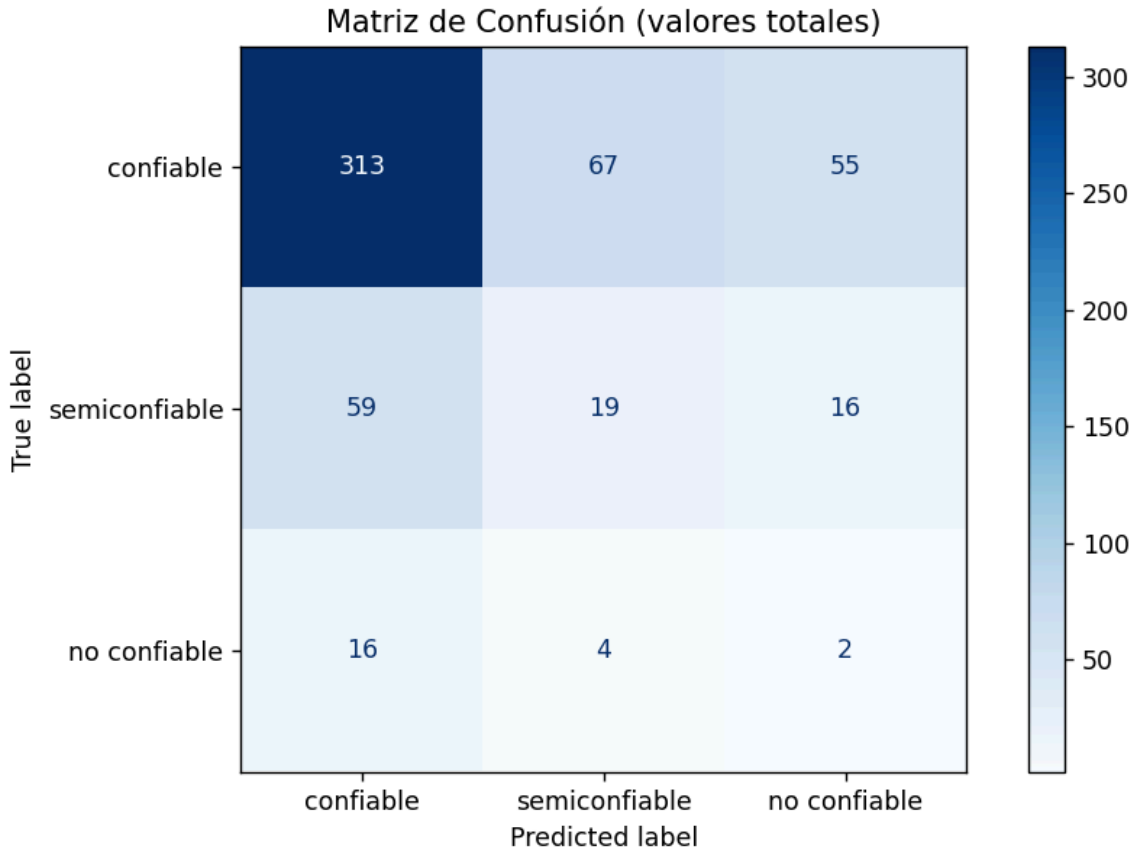

Ilustración 4.1 Matriz de confusión

Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real, o sea en términos prácticos nos permite ver qué tipos de aciertos y errores está teniendo nuestro modelo a la hora de pasar por el proceso de aprendizaje con los datos.

En la matriz de confusión presentada, podemos ver el rendimiento del sistema para las tres clases: "confiable," "semiconfiable," y "no confiable":

- El modelo predijo correctamente 313 de los 435 casos etiquetados como "confiable," mientras que clasificó incorrectamente 67 como "semiconfiable" y 55 como "no confiable."
- Para la clase "semiconfiable," solo 19 de las 94 instancias fueron correctamente clasificadas, con 59 casos clasificados erróneamente como "confiable" y 16 como "no confiable."
- Finalmente, en la clase "no confiable," 2 de las 22 instancias fueron clasificadas correctamente, mientras que 4 fueron clasificadas como "semiconfiable" y 16 como “confiable”.

Para generar la matriz de confusión en nuestro análisis, hemos utilizado el siguiente fragmento de código, el cual detallo a continuación:

```python
import pandas as pd
from sklearn.metrics import classification_report, confusion_matrix,
ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Leer el archivo CSV con las etiquetas verdaderas
real_df = pd.read_csv('C:/Users/elena/Desktop/trial_sin_orden.csv')
y_true = real_df['Reliability_Label'].tolist()  # Extraer la columna
'Reliability_Label'

# Leer el archivo CSV con las etiquetas predichas
predict_df = pd.read_csv('C:/Users/elena/Desktop/modelos/3_cot/archivo.csv')
y_pred = predict_df['Reliability_Label'].tolist()  # Extraer la columna
'Reliability_Label'

# Definir los nombres de las clases
target_names = ['confiable', 'semiconfiable', 'no confiable']

# Generar el reporte de clasificación
report = classification_report(y_true, y_pred, target_names=target_names)
print(report)

# Generar la matriz de confusión
conf_matrix = confusion_matrix(y_true, y_pred, labels=target_names)

# Visualizar la matriz de confusión
disp = ConfusionMatrixDisplay(confusion_matrix=conf_matrix,
display_labels=target_names)
disp.plot(cmap='Blues', values_format='d')  # 'd' para mostrar enteros
plt.title('Matriz de Confusión (valores totales)')
plt.show()
```

Este código desarrollado en Python, carga las etiquetas verdaderas y predichas desde dos los archivos CSV, genera un reporte de clasificación para evaluar el modelo que muestra las principales métricas de evaluación de un sistema de clasificación, incluyendo la precisión, la cobertura (*recall*), el *F1-score* y el soporte para cada clase. Luego crea y muestra la matriz de confusión para visualizar el rendimiento del modelo en términos de las clases aplicando un mapa de colores azul y mostrando los valores como enteros, lo que me permite analizar la precisión del modelo y las confusiones entre las clases.

Para detectar patrones sobre cómo este modelo ha clasificado el texto en cada una de las clases de fiabilidad (“confiable”, “semiconfiable” y “no confiable”), haremos un análisis de 10 ejemplos de noticias seleccionados manualmente en los que el modelo pudo haber cometido errores.

4.2.2. Detección de patrones erróneos en la clasificación

La clasificación de fiabilidad mostrada en la anterior matriz de confusión indica que el modelo no ha clasificado las categorías correctamente. Esto puede deberse a que, como el modelo está previamente entrenado, podría estar extrayendo información errónea o sesgada de ese entrenamiento previo. No sabemos con qué datos se ha entrenado exactamente, lo que podría estar causando que algunas definiciones de clases, como "semiconfiable", sean más difusas.

Si los datos de entrenamiento originales contienen ejemplos ambiguos o mal etiquetados, el modelo puede aprender patrones poco precisos, lo que afecta su capacidad para distinguir adecuadamente entre las clases, especialmente en las categorías menos representadas o con límites menos claros. Por ejemplo, si el modelo se encontró con noticias etiquetadas como "semiconfiable" que realmente eran confiables o no confiables, al proporcionarle ejemplos del archivo de entrenamiento (5w1h_subtask_2_train.json), esto puede haber creado confusión en su capacidad de clasificación.

Para detectar estos posibles errores, analizaremos varios ejemplos de noticias para cada categoría de confiabilidad (“confiable”, ”semiconfiable” y “no confiable”).

- **confiable:**
    - **- Noticia:** "Tras encerrar en prisión al líder de la oposición, Rusia retoma la campaña internacional a favor de la Sputnik V."
      - **Valor real:** confiable
      - **Valor predicho:** semiconfiable
      - **Tag_Text:** “la campaña internacional a favor de la Sputnik V”
      - **Análisis:** Aunque la noticia parece tratar sobre eventos recientes, el sistema puede haber cometido un error al clasificarla como

semiconfiable debido al contexto político y a la falta de detalles específicos sobre el impacto de las acciones mencionadas.

- **- Noticia:** "La acrilamida está presente en nuestra dieta desde que el hombre empezó a cocinar los alimentos."
  - **Valor real:** confiable
  - **Valor predicho:** semiconfiable
  - **Tag_Text:** “en nuestra dieta”
  - **Análisis:** La noticia afirma un hecho sobre la acrilamida y su presencia en la dieta desde tiempos antiguos. La confusión en la clasificación puede deberse a la forma en la que el modelo interpretó el contexto o la fuente.

- **semiconfiable:**
  - **- Noticia**: "Los niños y adolescentes tienen un sistema inmunológico extremadamente activo y adaptable y necesitan una interacción constante con el microbioma de la tierra."
    - **Valor real**: semiconfiable
    - **Valor predicho**: confiable
    - **Tag_Text**: “los niños”
    - **Análisis**: Aunque la noticia proporciona información sobre el sistema inmunológico, la afirmación puede parecer exagerada o sin suficiente respaldo científico, lo que justifica una clasificación como semiconfiable.

  - **- Noticia**: "A pesar de los deseos de la población por dejar atrás ya este ‘annus horribilis’, la realidad dice que esto, cambie el año o no, está lejos de terminar."
    - **Valor real**: semiconfiable
    - **Valor predicho**: no confiable
    - **Tag_Text**: “la realidad”
    - **Análisis**: La noticia presenta una visión pesimista sobre el estado actual, pero el contenido es una opinión general y no una afirmación concreta. El sistema pudo haber interpretado la falta de evidencia y la naturaleza subjetiva de la afirmación como “no confiable”.

- **no confiable:**
    - **- Noticia**: “Cuando dentro de diez años la demencia aumente exponencialmente y las generaciones más jóvenes no puedan alcanzar su potencial innato, no servirá de nada decir que no necesitábamos las mascarillas.”
      - **Valor real**: no confiable
      - **Valor predicho**: confiable
      - **Tag_Text**: “dentro de diez años”
      - **Análisis**: La noticia hace una afirmación general que puede haber llevado al modelo a clasificarlo incorrectamente como “confiable” ya que un futuro hipotético puede proporcionar evidencia concreta o contexto adicional, por ello el sistema ha predicho el texto como “confiable”.

    - **- Noticia**: “Todo ser humano debe poder decidir libremente si quiere llevar una mascarilla absolutamente ineficaz para protegerse de un virus.”
      - **Valor real**: no confiable
      - **Valor predicho**: semiconfiable
      - **Tag_Text**: “si quiere llevar una mascarilla absolutamente ineficaz”
      - **Análisis**: Esta noticia presenta una opinión personal sobre el uso de mascarillas. El modelo puede haber interpretado esta noticia como “semiconfiable” debido al lenguaje subjetivo.

Se debe recordar que el conjunto de datos de entrenamiento con el que hemos trabajado, contiene 6.934 noticias, lo que, aunque es razonable, puede no ser suficiente para capturar la complejidad de todas las clases de forma equitativa.

4.3. Comparación de resultados obtenidos por los demás participantes en la Tarea FLARES

Como comentamos al principio del proyecto, hemos participado en la tarea FLARES que tiene como objetivo la detección precisa de la fiabilidad en las noticias escritas en español. Esta tarea se describe en la plataforma *Kaggle* donde podemos

ver la descripción general de la tarea, obtener los conjuntos de datos, la tabla de posiciones y un foro donde los participantes pueden compartir ideas y resolver dudas con otros miembros de la competencia, entre otros.

Para comparar nuestros resultados con los de otros participantes, consultamos la tabla de posiciones, que muestra en tiempo real el rendimiento de cada uno según las métricas de evaluación[27]. A continuación, se muestra una ilustración de dicha tabla con los resultados finales:

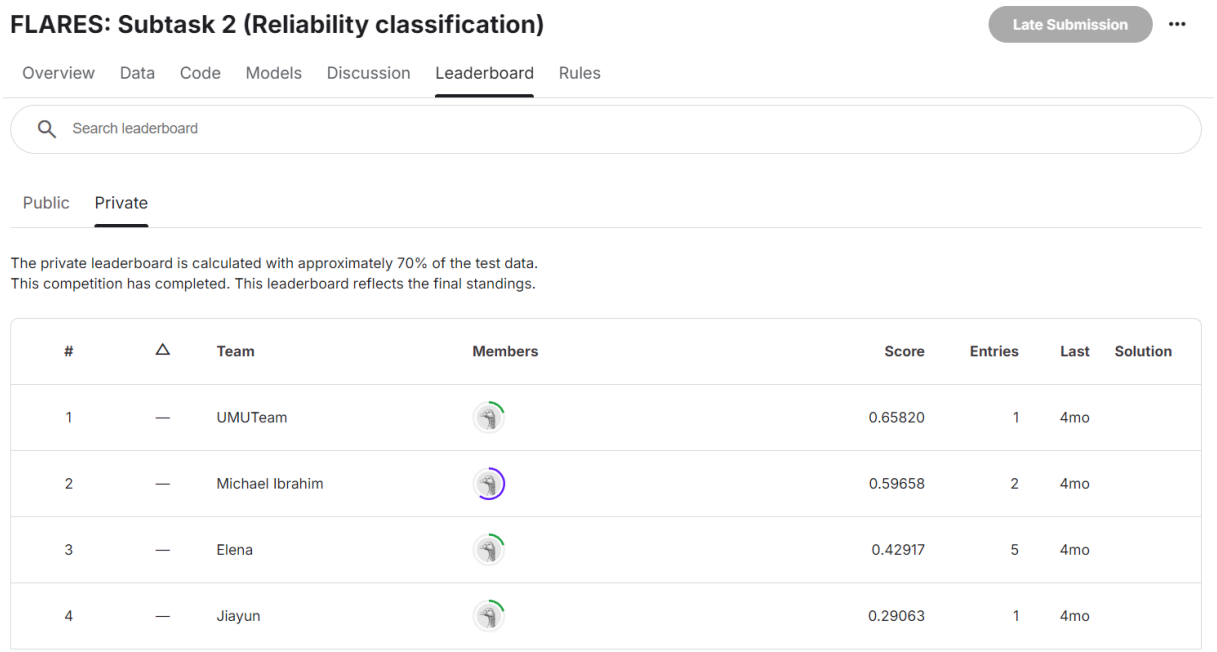

Ilustración 4.2 Tabla de posiciones de los participantes en la tarea FLARES

Fuente: Kaggle[27]

Es importante mencionar que la tarea comenzó el 21 de marzo y terminó el 24 de mayo de este mismo año. Finalmente por falta de tiempo y conocimiento sobre el área, no pude obtener el mejor rendimiento del sistema para este rango de fechas. Aún así podremos comparar los resultados finalmente obtenidos en este proyecto con los resultados de los demás equipos ya que, aunque la competición FLARES esté cerrada, aún continúa en curso para el envío de soluciones.

En la tabla de clasificación, se muestran dos apartados: público y privado, cada uno con su propia tabla de posiciones. La primera tabla corresponde a la sección pública en el que los resultados se basan en aproximadamente el 30% de los datos de prueba, los cuales son visibles para todos los participantes. Los

resultados finales se determinarán con el 70% restante, lo que pudo alterar el orden final de los equipos.

En esta fase preliminar, *UMUTeam* lidera con una puntuación de ***0.63757***, *Michael Ibrahim* sigue en segundo lugar con una puntuación de 0.58805. Por último, *Jiayun* se encuentra en el cuarto lugar con una puntuación de 0.30585. En esta sección, obtuvimos un resultado de 0.48168.

La sección privada de la tabla de clasificación que refleja los resultados finales de la competencia, se utilizó sobre aproximadamente el 70% de los datos de prueba. Esta puntuación no estuvo disponible hasta la fecha final de la tarea, lo que indica que el resultado final obtenido es el que se registró al terminar la competencia. *UMUTeam* mantuvo el primer lugar con una puntuación final de 0.65820. *Michael Ibrahim* ocupó el segundo lugar con una puntuación de 0.59658. Quedamos en tercer lugar con una puntuación de 0.42917. Y *Jiayun*, con una puntuación final de 0.29063 se situó en el cuarto lugar.

Si comparamos los resultados de los participantes con los obtenidos finalmente en este proyecto, donde el resultado más alto fue de **0.52730**, se observa que los líderes de la competencia superaron significativamente nuestro mejor desempeño. Es importante señalar que en nuestro proyecto solo se utilizaron los modelos generativos *Gemini* y *ChatGPT* para predecir los resultados, mientras que otros participantes probablemente usaron diferentes enfoques o modelos. Esto resalta que, aunque nuestro rendimiento es competitivo, hay un potencial considerable para mejorar mediante la optimización de modelos y técnicas de ajuste. Las diferencias en las puntuaciones finales subrayan la importancia de las estrategias de modelado y la calidad del ajuste del modelo, lo que puede ofrecer oportunidades para mejorar en futuras competiciones.

4.4. Análisis de errores de ejecución

Como es inevitable en proyectos de esta complejidad, se han presentado errores de ejecución. Estos problemas son normales cuando se está mejorando y

desarrollando modelos avanzados. Enfrentarlos y solucionarlos es clave para asegurarnos de que el proyecto tenga un buen resultado.

- El primer error ocurrió cuando la cuota asignada para el uso de la API de OpenAI se excedía.

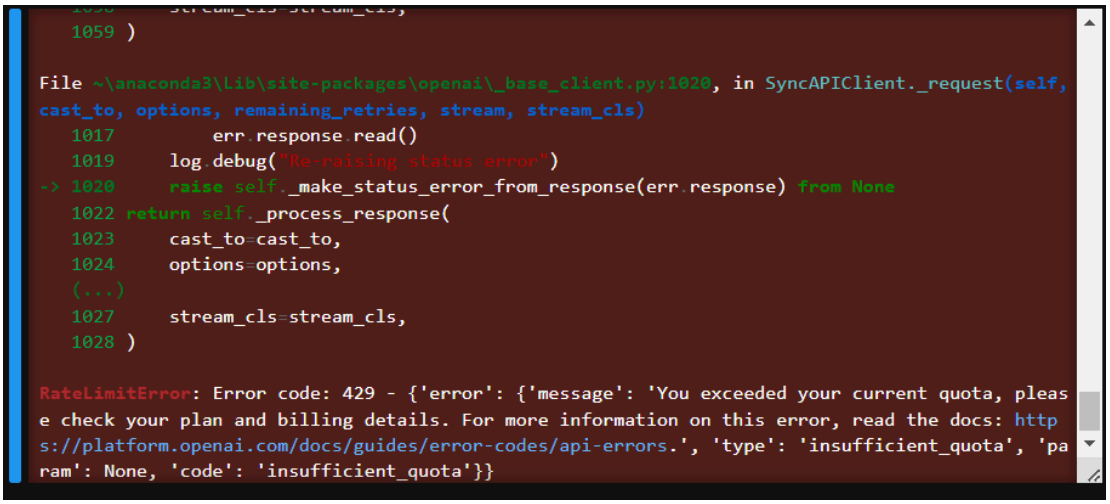

Ilustración 4.3 Primer error de ejecución

En este caso, el código intentó realizar una solicitud a la API de OpenAI, pero recibió un código de error 429 indicando que se ha superado el límite de uso permitido en el plan actual. Como resultado, la solicitud no se pudo completar y el flujo de trabajo fue interrumpido. Para resolver esto, recurrimos al plan de suscripción de pago por tokens y así aumentamos el número de solicitudes realizadas.

- Otro error surgió cuando el sistema encontró un carácter no reconocido, lo que llevó a fallos significativos en la interpretación del texto. A pesar de que especificamos al modelo que debía limitarse a usar predecir los tres valores permitidos de fiabilidad, el modelo no cumplió con esta restricción en su totalidad.

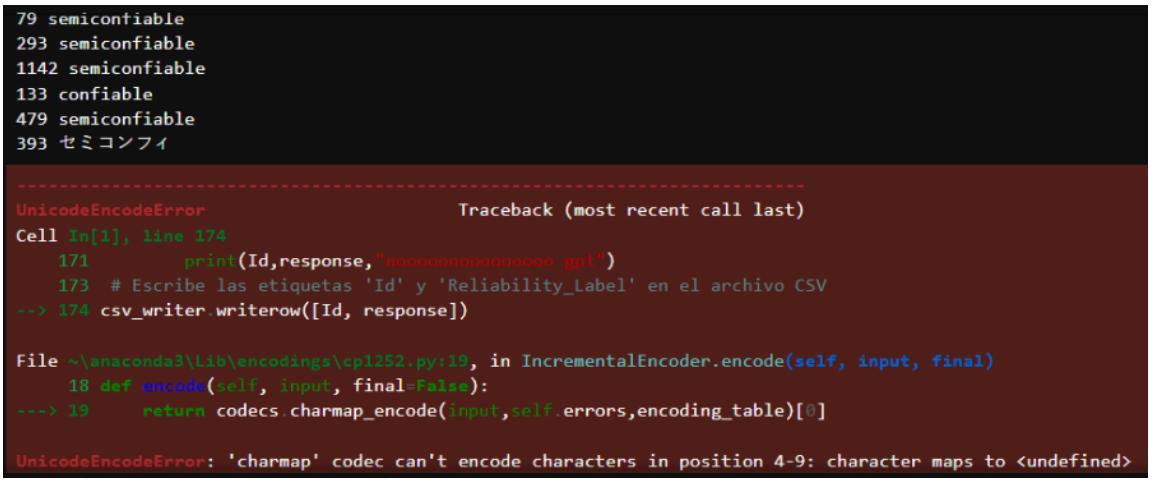

Ilustración 4.4 Segundo error de ejecución

Este error detiene la ejecución del modelo y es importante que lo solucionemos. Para ello, hemos ejecutado el código de nuevo ya que ha surgido en 2 de las 52 ejecuciones del código, lo que representa aproximadamente el 3.8% de las veces.

- Uno de los problemas más comunes ocurrió cuando el modelo, en algunas ocasiones, estaba generando predicciones que no se ajustaban a los tres valores de fiabilidad esperados: "confiable", "semiconfiable" y "no confiable". En lugar de limitarse a estos tres resultados, el modelo estaba produciendo salidas inesperadas o adicionales, lo cual no es aceptable para nuestro sistema de clasificación.

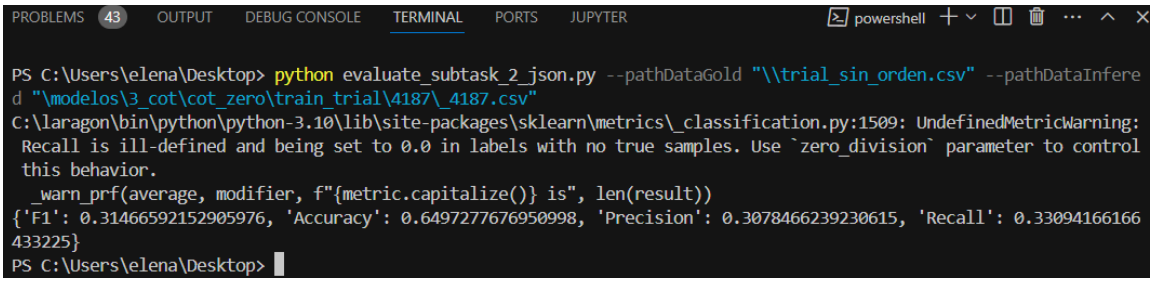

Ilustración 4.5 Tercer error de ejecución

Para abordar este problema, implementamos un proceso de post-procesamiento de los datos. Este ajuste adicional se encarga de revisar y corregir las predicciones del modelo para garantizar que todas se clasifiquen correctamente dentro de las tres categorías definidas. Este enfoque lo comentaremos en el siguiente apartado.

5. CONCLUSIONES

En este capítulo, resumimos los principales resultados obtenidos, exploramos ideas para futuros trabajos, proponemos posibles mejoras y detallamos el artículo en el que se presentará el estudio de este proyecto.

5.1. Conclusiones sobre resultados finales

Primero, evaluaremos si se han alcanzado los objetivos establecidos inicialmente.

El primer objetivo trata sobre estudiar técnicas de PLN para procesar la información de un texto. Esto se ha cumplido ya que se han explorado las diferentes técnicas de PLN, como la tokenización, lematización y stemming, eliminación de stopwords, normalización y la codificación de texto. Estas técnicas han permitido transformar y estructurar el texto de manera eficiente, facilitando el análisis y la extracción de información relevante para cada una de las noticias del dataset.

El segundo objetivo plantea estudiar las distintas aproximaciones que hay actualmente para la detección de fiabilidad en textos usando PLN. También hemos cumplido con este objetivo ya que se ha explicado cómo las técnicas de PLN han ayudado a resolver la detección de fiabilidad en las noticias. Primero, se realiza un entrenamiento del modelo con los datos etiquetados, luego se lleva a cabo un análisis del texto para extraer características relevantes. Después, el modelo genera predicciones sobre la fiabilidad de las noticias y, por último, clasifica el texto según la fiabilidad de la noticia.

El tercer objetivo propone analizar el conjunto de datos proporcionado en la competición FLARES del workshop IberLEF 2024. Este objetivo se ha cumplido mediante una evaluación detallada de los tres conjuntos de datos proporcionados, en la que se ha examinado su estructura y relevancia. El análisis ha incluido la distribución de las clases, el tamaño del corpus y la correlación que tienen entre sí.

El cuarto objetivo idea realizar un estudio sobre las tres técnicas de ingeniería de prompts más utilizadas, explorando su uso y aplicación en la detección de la fiabilidad de noticias en español. Este objetivo se ha cumplido debido a que se analizaron estas técnicas exhaustivamente. Desde su definición hasta su aplicación en el contexto de la detección de fiabilidad de noticias. Se ha abordado cómo cada técnica puede ser utilizada para mejorar la precisión en la clasificación de la fiabilidad de las noticias en español, evaluando sus limitaciones y la eficacia de cada una en diferentes escenarios.

El quinto objetivo que menciona seleccionar un entorno de desarrollo adecuado y diseñar un prompt clasificador automático para evaluar la fiabilidad de noticias en español utilizando modelos generativos como Gemini y ChatGPT, también se ha alcanzado como lo hemos podido observar en el apartado 3 con el desarrollo del sistema de clasificación. Se optó por *Jupyter Notebook* como entorno de desarrollo para implementar y probar el clasificador. La descripción de la tarea de “Clasificación de fiabilidad” en Kaggle se utilizó como base para contextualizar y ajustar el modelo, asegurando así que el clasificador cumpliera con los requisitos específicos y las mejores prácticas del campo.

El sexto objetivo trata sobre evaluar los resultados y verificar la precisión del sistema de clasificación mediante un análisis de errores y el uso de una matriz de confusión. Este objetivo también se ha completado ya que realizamos un análisis de errores y utilizamos una matriz de confusión para evaluar la precisión del sistema en cada una de las ejecuciones.

Finalmente, el último objetivo planteado en redactar una memoria del proyecto ha sido completamente logrado. Este documento detalla todo el proceso de desarrollo del proyecto desde la investigación inicial y estudio de las técnicas de prompt, hasta la implementación y evaluación del sistema. La memoria incluye un análisis exhaustivo de los resultados obtenidos, así como reflexiones sobre los desafíos enfrentados y las posibles mejoras que podrían implementarse en el futuro. Además, sirve como un registro completo del esfuerzo realizado y una guía para futuras investigaciones y desarrollos en el campo de la clasificación de fiabilidad de noticias utilizando modelos generativos..

Después de realizar numerosas pruebas con diferentes prompts desarrollados, se concluye que *Gemini* no ha mostrado un rendimiento especialmente eficaz para la tarea de detección de fiabilidad en noticias. El mejor valor obtenido ha sido de **0.52730** en el *F1-score* utilizando la técnica de prompt con pocas muestras o “few-shot”. Esto sugiere que el modelo presenta limitaciones en su capacidad para equilibrar precisión y exhaustividad en esta tarea específica.

También se ha observado que, aunque un sistema pueda obtener buenos resultados con el conjunto de datos de prueba, el rendimiento puede variar significativamente al predecir etiquetas en el conjunto de datos de test. Esto puede resultar en una disminución o aumento en las métricas de evaluación, lo que destaca la importancia de una evaluación continua y exhaustiva para garantizar la robustez y generalización del sistema en escenarios reales.

Para concluir, este proyecto me ha dado una gran perspectiva sobre la ingeniería de prompts, incluyendo sus técnicas, aplicaciones y usos prácticos. Además, he aprendido a aplicar métricas de evaluación para analizar el rendimiento de sistemas clasificadores, lo que ha sido fundamental para ajustar y perfeccionar el sistema. Trabajar con la herramienta Kaggle no solo me ha permitido acceder a los conjuntos de datos proporcionados para el desarrollo del proyecto sino también explorar la plataforma, que ofrece desafíos de ciencia de datos, competiciones, recursos educativos y la oportunidad de aprender de una comunidad activa de expertos.

Todo esto me ha dado una perspectiva más clara y práctica en este campo, preparándome mejor para enfrentar futuros desafíos en el PLN e IA con una base más sólida.

5.2. Posibles mejoras

Dado que este proyecto se ha centrado en investigar el rendimiento del sistema y no se ha obtenido un resultado muy elevado, es importante considerar diversas estrategias de mejora. A continuación, se presentan tres enfoques que pueden ayudar a optimizar nuestro sistema: aumentar el conjunto de datos,

balancear las clases y entrenar con un modelo más avanzado como BERT. Estas técnicas pueden ofrecer nuevas perspectivas para mejorar la precisión y la robustez del sistema.

- **Aumento del conjunto de datos:** un conjunto de datos más amplio permite que el modelo entienda mejor las distintas maneras en que se puede expresar un concepto. Esto es especialmente útil con prompts, ya que el modelo se entrena con una variedad más amplia de ejemplos y contextos. Podemos lograrlo creando variaciones del prompt a través de técnicas como la generación de sinónimos o reescritura de frases. Otra estrategia es la traducción y retrotraducción, que consiste en traducir una oración a otro idioma y luego devolverla al idioma original, generando nuevas oraciones con estructuras diferentes. Este enfoque no sólo amplía el conjunto de entrenamiento, sino que también ayuda a Gemini a aprender a manejar la variabilidad en el lenguaje, lo que puede mejorar su capacidad para producir respuestas más precisas y, en consecuencia, elevar el F1-score.

- **Balanceo de clases:** para evitar que el modelo favorezca ciertas respuestas solemos recurrir a un balance en los datos de entrenamiento. En nuestro conjunto de datos de entrenamiento, las etiquetas de fiabilidad estaban muy desbalanceadas, con muchas más etiquetas "confiables" en comparación con las demás. Para mitigar este problema, seleccionamos un tamaño fijo para cada una de las etiquetas de confiabilidad. Realizar un balance de los datos utilizando técnicas como el sobremuestreo de las clases minoritarias o el submuestreo de las mayoritarias podrían haber llevado al modelo a predecir con mayor precisión. El sobremuestreo aumenta los ejemplos de las clases menos representadas, mientras que el submuestreo reduce los ejemplos de las clases más representadas, ayudando a evitar el sesgo y permitiendo que el modelo aprenda de manera más equilibrada.

- **Entrenar con otro modelo:** considerar entrenar con otro modelo como BERT puede ser útil como un punto de comparación o para generar un conjunto de datos adicional. BERT (Bidirectional Encoder Representations from Transformers) es un modelo de lenguaje desarrollado por Google que se basa en la arquitectura de transformadores. A diferencia de los modelos como

Gemini y chatGPT que leen el texto de manera secuencial, BERT procesa el texto en ambos sentidos, permitiendo una mejor comprensión del contexto y las relaciones entre palabras. Sus variantes están previamente entrenadas en una amplia variedad de textos y pueden ser ajustados específicamente para tareas de generación de prompts. Podemos utilizar BERT para desarrollar un conjunto de datos etiquetado con alta precisión que luego se utilice para refinar el prompt de Gemini. Esto puede ayudar a maximizar el F1-score de manera significativa.

5.3. Trabajos futuros

Aunque este proyecto ha llegado a su fin, existen numerosas formas de seguir desarrollando y perfeccionando el sistema más allá de las mejoras específicas realizadas en el proyecto.

En primer lugar, es fundamental re-entrenar el modelo utilizando una base de conocimientos más amplia. Aunque Gemini tiene acceso a una gran cantidad de datos generales, es posible que no disponga de suficientes ejemplos específicos para el contexto de detección de fiabilidad en las noticias en español. Por lo tanto, sería beneficioso ampliar y enriquecer el conjunto de datos con ejemplos más variados y representativos.

Otro aspecto clave es el proceso de fine-tuning, que implica ajustar el modelo previamente entrenado con nuestros datos específicos. Aunque ya contamos con un conjunto de datos específicos en el contexto, existe la posibilidad de que haya un modelo con un corpus más centrado en el ámbito de fiabilidad en las noticias y nos ofrezca mejores resultados.

Por último, ajustar los hiperparámetros del modelo podría proporcionar mejoras adicionales en el rendimiento. Aunque ya hemos probado cambiar los ajustes pero no hemos obtenido resultados significativos, es posible que en otros conjuntos de datos sea más efectivo.

Para concluir el desarrollo de este proyecto, se redactará un artículo exhaustivo sobre la detección de fiabilidad en las noticias. Este artículo reflejará de

manera detallada todo lo investigado y analizado durante el proyecto, incluyendo las metodologías empleadas, los modelos utilizados, y los resultados obtenidos. En él, abordaremos las técnicas de clasificación utilizadas, las métricas de evaluación y la aplicabilidad práctica de nuestras conclusiones, ofreciendo una visión completa y fundamentada sobre cómo mejorar la veracidad de la información en los medios de comunicación.

5.4. Satisfacción personal

Completar este TFG ha sido una experiencia gratificante ya que no solo me ha permitido consolidar los conocimientos adquiridos a lo largo de mi carrera, sino también explorar nuevas áreas de interés y superar retos personales y académicos.

Haber desarrollado una investigación que refleja mi esfuerzo, dedicación y curiosidad intelectual ha sido una experiencia muy satisfactoria, ya que he logrado cumplir todos los objetivos que establecí al comienzo del trabajo, obteniendo resultados finales prometedores.

La elaboración del proyecto me ha llevado más tiempo del que esperaba pero considero que los resultados finales y la adaptación realizada me han permitido obtener un gran conocimiento sobre cómo se comportan los modelos generativos según la redacción del prompt.

En lo referente a la parte de programación, el desarrollo del sistema de detección de fiabilidad de noticias ha sido todo un desafío. Trabajar con un conjunto de datos desconocido y crear un sistema desde cero sin experiencia previa no es tarea fácil; tuve que dedicar tiempo a comprender el funcionamiento del PLN antes de poder diseñarlo. Gracias a las aportaciones realizadas por mi tutora Salud María, pude elaborar las modificaciones requeridas para concluir el proyecto.

Este proceso me ha brindado la oportunidad de crecer, aprender y aplicar de manera práctica lo que he estudiado, reforzando mi confianza en mis capacidades. Al mirar hacia atrás, siento orgullo por el trabajo realizado y una gran satisfacción personal por haber alcanzado este importante logro académico.

6. ANEXOS

Se presentan dos anexos en los que se incluye una descripción detallada de todos los prompts desarrollados para cada técnica y los manuales tanto de instalación como de usuario.

6.1. Anexo A: Descripción de prompts

A continuación, se presenta una tabla para cada técnica de prompt utilizada: “zero-shot”, “few-shot”, “CoT-zero-shot” y “CoT-few-shot”.

6.1.1. Técnica “zero-shot”

Modelos	Descripción del prompt	F1-score
Modelo 1	""" ### Instrucción ### Evalúa la fiabilidad de un fragmento de texto (`Tag_Text`) en relación con su categoría 5W1H (`5W1H_Label`). Debes clasificar el fragmento en una de las siguientes categorías de fiabilidad: "confiable", "semiconfiable" o "no confiable". **Descripción de las Etiquetas:** - **Text**: La noticia completa de la que el fragmento es parte. - **Tag_Text**: Un fragmento específico de la noticia que requiere evaluación. - **5W1H_Label**: La categoría que describe el tipo de información en el fragmento: - **WHAT (Qué)**: Hechos, objetos o detalles específicos. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Fechas, tiempos o periodos indicados. - **WHERE (Dónde)**: Lugares o ubicaciones mencionadas. - **WHY (Por qué)**: Razones o causas dadas. - **HOW (Cómo)**: Métodos o procedimientos descritos. **Criterios de Evaluación:** 1. **Precisión**: Verifica si el fragmento es claro, específico y se puede corroborar. La precisión refleja la exactitud de los detalles. 2. **Objetividad**: Asegúrate de que el fragmento es imparcial y libre de opiniones o sesgos. La objetividad implica neutralidad en la información presentada. **Categorías de Fiabilidad:** - **confiable**: El fragmento es preciso y objetivo. - **semiconfiable**: El fragmento cumple solo uno de los dos criterios: precisión u objetividad. - **no confiable**: El fragmento no cumple con los criterios de precisión ni de objetividad. **Instrucciones Finales:** - Usa "semiconfiable" solo cuando es evidente que el fragmento cumple uno de los dos	**0.45875**

	criterios. - Responde con una sola palabra: "confiable", "semiconfiable" o "no confiable". Evita añadir cualquier puntuación extra. """	
Modelo 2	""" ### Instrucción ### Tu tarea es evaluar la fiabilidad de un fragmento de texto (`Tag_Text`) en función de su categoría 5W1H (`5W1H_Label`). Debes clasificar la fiabilidad en una de las siguientes tres categorías: "confiable", "semiconfiable" o "no confiable". **Detalles de las Etiquetas:** - **Text**: La noticia completa de la cual el fragmento es parte. - **Tag_Text**: Un fragmento de la noticia que debes evaluar. - **5W1H_Label**: La categoría de la etiqueta que describe el tipo de información contenida en el fragmento: - **WHAT (Qué)**: Información sobre hechos, objetos o detalles específicos. - **WHO (Quién)**: Información sobre personas o entidades involucradas. - **WHEN (Cuándo)**: Información sobre fechas, tiempos o periodos. - **WHERE (Dónde)**: Información sobre lugares o ubicaciones. - **WHY (Por qué)**: Información sobre razones o causas. - **HOW (Cómo)**: Información sobre métodos o procedimientos. **Criterios de Evaluación:** 1. **Precisión**: Verifica si el contenido del fragmento es claro y puede ser corroborado. La precisión se refiere a la exactitud de los detalles proporcionados. 2. **Objetividad**: Evalúa si el contenido es imparcial y libre de sesgos. La objetividad implica que el texto presenta la información de manera neutral y sin opiniones personales. **Categorías de Fiabilidad:** - **confiable**: El fragmento es tanto preciso como objetivo. - **semiconfiable**: El fragmento cumple solo uno de los dos criterios (precisión u objetividad). - **no confiable**: El fragmento no cumple con los criterios de precisión ni de objetividad. **Instrucciones Adicionales:** - Usa "semiconfiable" solo cuando estés seguro de que el fragmento cumple solo uno de los criterios. - Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable". No incluyas puntuación adicional. """	**0.44667**

Modelo 3	""" ### Instrucción ### Tu tarea es evaluar la fiabilidad de un fragmento de texto (`Tag_Text`) en relación con su categoría 5W1H (`5W1H_Label`). La fiabilidad se clasifica en tres categorías: "confiable", "semiconfiable" y "no confiable". **Descripción de las etiquetas:** - **Text**: La noticia completa. - **Tag_Text**: Un fragmento de la noticia que se debe evaluar. - **5W1H_Label**: Categoría que describe el fragmento de texto según el acrónimo 5W1H: - **WHAT (Qué)**: Hechos, objetos o elementos concretos. - **WHO (Quién)**: Personas o entidades. - **WHEN (Cuándo)**: Fechas o tiempos. - **WHERE (Dónde)**: Lugares o direcciones. - **WHY (Por qué)**: Razones o causas. - **HOW (Cómo)**: Métodos o procesos. **Criterios para la clasificación:** 1. **Precisión**: La información debe ser específica y verificable. Ejemplos de alta precisión incluyen fechas exactas o entidades conocidas. 2. **Objetividad**: El contenido debe ser neutral y libre de opiniones personales o sesgos. **Categorías de fiabilidad:** - **confiable**: La información es tanto precisa como objetiva. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). - **no confiable**: La información no es precisa ni objetiva. **Instrucciones adicionales:** - Usa "semiconfiable" solo si estás seguro de que la noticia cumple únicamente uno de los dos criterios. - Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable". No utilices puntuación adicional. """	**0.43294**

Modelo 4	""" ### Instrucción ### Evalúa la fiabilidad de un fragmento de texto (`Tag_Text`) en relación con su categoría 5W1H (`5W1H_Label`). Clasifica la fiabilidad en una de las siguientes tres categorías: "confiable", "semiconfiable" o "no confiable". **Definición de las Etiquetas:** - **Text**: La noticia completa de la que se extrajo el fragmento. - **Tag_Text**: El fragmento específico de la noticia que debes evaluar. - **5W1H_Label**: La categoría que describe el tipo de información contenida en el fragmento: - **WHAT (Qué)**: Detalles sobre hechos, objetos o eventos específicos. - **WHO (Quién)**: Información sobre personas o entidades mencionadas. - **WHEN (Cuándo)**: Fechas, horarios o periodos de tiempo mencionados. - **WHERE (Dónde)**: Información sobre lugares o ubicaciones específicas. - **WHY (Por qué)**: Razones, causas o motivaciones indicadas. - **HOW (Cómo)**: Métodos, procesos o procedimientos descritos. **Criterios de Evaluación:** 1. **Precisión**: Evalúa si el fragmento proporciona información clara, específica y verificable. La precisión se refiere a la exactitud de los detalles presentados. 2. **Objetividad**: Considera si el fragmento es imparcial y libre de sesgos. La objetividad implica que el contenido se presenta de manera neutral, sin opiniones personales. **Clasificación de Fiabilidad:** - **confiable**: El fragmento es tanto preciso como objetivo. - **semiconfiable**: El fragmento cumple solo uno de los dos criterios, ya sea precisión u objetividad. - **no confiable**: El fragmento no cumple con ninguno de los dos criterios. **Instrucciones Finales:** - Usa "semiconfiable" únicamente cuando el fragmento cumpla claramente solo uno de los dos criterios. - Responde exclusivamente con una de las siguientes palabras en minúsculas: "confiable", "semiconfiable" o "no confiable". No incluyas puntuación adicional ni otras palabras. """	**0.42840**

Modelo 5	""" ### Instrucción ### Evalúa la fiabilidad de un fragmento de texto (`Tag_Text`) basado en su categoría 5W1H (`5W1H_Label`). Clasifica la fiabilidad del fragmento en una de las siguientes categorías: "confiable", "semiconfiable" o "no confiable". **Definición de las Etiquetas:** - **Text**: La noticia completa de la que proviene el fragmento. - **Tag_Text**: El fragmento específico de la noticia que necesitas evaluar. - **5W1H_Label**: La categoría que describe la naturaleza del fragmento: - **WHAT (Qué)**: Hechos, objetos o detalles específicos. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Fechas, horas o periodos de tiempo. - **WHERE (Dónde)**: Lugares o ubicaciones mencionadas. - **WHY (Por qué)**: Razones, causas o motivaciones indicadas. - **HOW (Cómo)**: Métodos, procesos o procedimientos descritos. **Criterios de Evaluación:** 1. **Precisión**: Determina si el fragmento es específico y verificable. La precisión se refiere a la exactitud y concreción de los detalles presentados. 2. **Objetividad**: Evalúa si el fragmento es neutral y libre de sesgos. La objetividad implica que el texto se presente de manera imparcial, sin opiniones personales. **Clasificación de Fiabilidad:** - **confiable**: El fragmento cumple tanto con la precisión como con la objetividad. - **semiconfiable**: El fragmento cumple con uno de los dos criterios, ya sea precisión u objetividad. - **no confiable**: El fragmento no cumple con ninguno de los dos criterios. **Instrucciones Finales:** - Usa "semiconfiable" solo cuando el fragmento cumpla claramente solo uno de los dos criterios. - Responde exclusivamente con una de las siguientes palabras en minúsculas: "confiable", "semiconfiable" o "no confiable". Evita agregar cualquier otra puntuación. """	**0.42772**

Modelo 6	""" ### Instrucción ### Evalúa la fiabilidad de un fragmento de texto (`Tag_Text`) basado en su categoría 5W1H (`5W1H_Label`). Clasifica la fiabilidad del fragmento en una de las siguientes categorías: "confiable", "semiconfiable" o "no confiable". **Definición de las Etiquetas:** - **Text**: La noticia completa de la que proviene el fragmento. - **Tag_Text**: El fragmento específico de la noticia que necesitas evaluar. - **5W1H_Label**: La categoría que describe la naturaleza del fragmento: - **WHAT (Qué)**: Hechos, objetos o detalles específicos. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Fechas, horas o periodos de tiempo. - **WHERE (Dónde)**: Lugares o ubicaciones mencionadas. - **WHY (Por qué)**: Razones, causas o motivaciones indicadas. - **HOW (Cómo)**: Métodos, procesos o procedimientos descritos. **Criterios de Evaluación:** 1. **Precisión**: Determina si el fragmento es específico y verificable. La precisión se refiere a la exactitud y concreción de los detalles presentados. 2. **Objetividad**: Evalúa si el fragmento es neutral y libre de sesgos. La objetividad implica que el texto se presente de manera imparcial, sin opiniones personales. **Clasificación de Fiabilidad:** - **confiable**: El fragmento cumple tanto con la precisión como con la objetividad. - **semiconfiable**: El fragmento cumple con uno de los dos criterios, ya sea precisión u objetividad. - **no confiable**: El fragmento no cumple con ninguno de los dos criterios. **Instrucciones Finales:** - Usa "semiconfiable" solo cuando el fragmento cumpla claramente solo uno de los dos criterios. - Responde exclusivamente con una de las siguientes palabras en minúsculas: "confiable", "semiconfiable" o "no confiable". Evita agregar cualquier otra puntuación. """	**0.41902**

Modelo 7	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia basada en la relación entre un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). Las categorías 5W1H son: - **WHAT (Qué)**: Hechos, objetos o elementos concretos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Tiempos o periodos mencionados. - **WHERE (Dónde)**: Lugares o direcciones específicas. - **WHY (Por qué)**: Razones o causas mencionadas. - **HOW (Cómo)**: Métodos o procedimientos detallados. Para cada noticia, considera lo siguiente: 1. **Precisión**: ¿La información es específica y verificable? Ejemplos de alta precisión incluyen fechas exactas o instituciones reconocidas. 2. **Objetividad**: ¿El texto es neutral y sin sesgo? Evalúa si el autor expresa opiniones personales o si el texto tiene una carga emocional. Basado en estas consideraciones, clasifica la noticia en una de las siguientes categorías de fiabilidad: - **confiable**: La información es precisa y objetiva. - **no confiable**: La información carece tanto de precisión como de objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Nota:** Evita usar "semiconfiable" a menos que estés seguro de que solo se cumple uno de los criterios. Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin puntos ni guiones. """	**0.41089**

Modelo 8	""" ### Instrucción ### Evalúa la fiabilidad de un fragmento de texto (`Tag_Text`) en relación con su categoría 5W1H (`5W1H_Label`). Debes clasificar la fiabilidad en una de las siguientes categorías: "confiable", "semiconfiable" o "no confiable". **Definición de Etiquetas:** - **Text**: El contenido completo de la noticia. - **Tag_Text**: Un fragmento específico del texto de la noticia que se evaluará. - **5W1H_Label**: Clasificación del fragmento según la categoría 5W1H: - **WHAT (Qué)**: Hechos, objetos, o detalles específicos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Fechas, tiempos o periodos indicados. - **WHERE (Dónde)**: Lugares o ubicaciones mencionadas. - **WHY (Por qué)**: Razones, causas o motivaciones. - **HOW (Cómo)**: Métodos, procedimientos o formas en que se realiza algo. **Criterios para la Evaluación de Fiabilidad:** 1. **Precisión del Contenido**: Determina si el fragmento es específico y puede verificarse con hechos claros y detallados. 2. **Objetividad del Contenido**: Considera si el fragmento se presenta de manera neutral, sin influencia de opiniones personales o sesgos. **Clasificación de Fiabilidad:** - **confiable**: El fragmento es preciso y objetivo. - **semiconfiable**: El fragmento cumple solo uno de los criterios, ya sea precisión u objetividad. - **no confiable**: El fragmento no cumple ni con el criterio de precisión ni con el de objetividad. **Instrucciones Finales:** - Utiliza "semiconfiable" únicamente si es evidente que el fragmento cumple con uno solo de los criterios. - Responde con una palabra en minúsculas: "confiable", "semiconfiable" o "no confiable". Evita el uso de cualquier puntuación adicional. """	**0.41001**

Modelo 9	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de un fragmento de texto (`Tag_Text`) basándote en su relación con una categoría 5W1H (`5W1H_Label`). La clasificación debe ser una de las siguientes: "confiable", "semiconfiable" o "no confiable". **Definiciones de las etiquetas:** - **Text**: La noticia completa. - **Tag_Text**: Un fragmento de la noticia que necesitas evaluar. - **5W1H_Label**: La categoría del fragmento de texto según el acrónimo 5W1H: - **WHAT (Qué)**: Hechos, objetos o elementos concretos. - **WHO (Quién)**: Personas o entidades. - **WHEN (Cuándo)**: Fechas, tiempos o períodos. - **WHERE (Dónde)**: Lugares o direcciones. - **WHY (Por qué)**: Razones, causas o motivaciones. - **HOW (Cómo)**: Métodos o procedimientos. **Criterios para la clasificación:** 1. **Precisión**: Evalúa si la información es específica y verificable. La alta precisión se manifiesta en detalles claros como fechas exactas o nombres de entidades reconocidas. 2. **Objetividad**: Determina si el contenido es neutral y libre de opiniones personales o sesgos. La objetividad implica que el texto no tenga juicios subjetivos ni carga emocional. **Categorías de fiabilidad:** - **confiable**: La información es tanto precisa como objetiva. - **semiconfiable**: La información cumple solo uno de los criterios (precisión u objetividad). - **no confiable**: La información no es precisa ni objetiva. **Instrucciones adicionales:** - Usa "semiconfiable" solo si estás seguro de que el fragmento cumple únicamente uno de los criterios. - Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable". No incluyas puntuación adicional. """	**0.39593**

Modelo 10	""" ### Instrucción ### Analiza la etiqueta Tag_Text y 5W1H_Label para predecir la fiabilidad de la etiqueta Tag_Text. La etiqueta Tag_Text es una porción continua de la etiqueta Text, está relacionada de manera significativa con la etiqueta 5W1H_Label y es una porción de la noticia de la etiqueta Text. El archivo de entrada contiene las siguientes etiquetas descriptivas: La etiqueta Text hace referencia a la noticia. La etiqueta Tag_Text hace referencia a una porción de texto de la etiqueta Text La etiqueta 5W1H_Label hace referencia a cada elemento del acrónimo. Las 5W representan lo que sería en ingles: "Qué", "Por qué", "Quién", "Cuándo" y "Dónde", mientras que la 1H representa "Cómo". La etiqueta Relibility_Label hace referencia a la fiabilidad de la noticia. Puede ser únicamente los valores: confiable y no confiable. Las categorías de etiquetas son las siguientes: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. WHEN (Cuándo): Momentos o periodos de tiempo especificados. WHERE (Dónde): Lugares, espacios geográficos o direcciones. WHY (Por qué): Causas, razones o motivaciones. HOW (Cómo): Maneras, métodos o procedimientos detallados. Hay 2 categorías las cuales son confiable y no confiable. La etiqueta "confiable" se utiliza cuando la información es tanto objetiva como precisa; la etiqueta "no confiable" cuando no cumple ni el criterio de objetividad ni el de precisión, y la etiqueta "semiconfiable" se utilizará cuando solo cumpla uno de los dos criterios. Se debe evitar escribir semiconfiable a no ser que estés realmente seguro de que no cumple un criterio al 100%. Te tienes que basar en: Escribe la etiqueta confiable, semiconfiable o no confiable asociada a la información de entrada, sin punto, en minúscula y sin guiones usando literalmente alguna de estas tres categorías. """	**0.36287**

6.1.2. Técnica “few-shot”

Modelos	Descripción del prompt	F1-score
Modelo 1	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia evaluando la relación entre un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). Las categorías 5W1H son las siguientes: - **WHAT (Qué)**: Hechos, objetos o elementos específicos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Fechas, tiempos o periodos mencionados. - **WHERE (Dónde)**: Lugares o direcciones específicas. - **WHY (Por qué)**: Razones, causas o motivaciones mencionadas. - **HOW (Cómo)**: Métodos, procesos o procedimientos detallados. Para cada noticia, analiza los siguientes aspectos: 1. **Precisión del contenido**: ¿Es la información específica, clara y verificable? La precisión implica que la información sea detallada y respaldada por evidencia. - Ejemplo de información precisa: "El viernes 19 de marzo" (confiable). - Ejemplo de información imprecisa: "Hace mucho tiempo" (no confiable). 2. **Objetividad del contenido**: ¿Es el texto neutral y libre de sesgos? La objetividad se refleja en la falta de opiniones personales, emociones o subjetividad. - Ejemplo de información objetiva: "Según el rector de la Universidad de Alicante" (confiable). - Ejemplo de información subjetiva: "En mi opinión" (no confiable). En base a estos criterios, clasifica la noticia en una de las siguientes categorías de fiabilidad: - **confiable**: La información es tanto precisa como objetiva. - **no confiable**: La información carece de precisión y objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Instrucciones adicionales**: - Usa "semiconfiable" solo si estás seguro de que solo se cumple uno de los dos criterios. - Escribe tu respuesta en minúsculas, sin puntos ni guiones: "confiable", "semiconfiable" o "no confiable". A continuación, se te presentan ejemplos concretos para guiarte en la clasificación: """	**0.46242**

Modelo 2	""" ### Instrucción ### Tu tarea es evaluar la fiabilidad de una noticia analizando la relación entre un fragmento de texto (`Tag_Text`) y su correspondiente categoría 5W1H (`5W1H_Label`). Las categorías 5W1H definen la naturaleza del contenido como sigue: - **WHAT (Qué)**: Hechos, objetos o elementos específicos mencionados. - **WHO (Quién)**: Personas o entidades implicadas. - **WHEN (Cuándo)**: Fechas, tiempos o periodos especificados. - **WHERE (Dónde)**: Lugares, ubicaciones geográficas o direcciones. - **WHY (Por qué)**: Motivos, razones o causas presentadas. - **HOW (Cómo)**: Métodos, procesos o procedimientos descritos. Para cada fragmento de texto, considera dos aspectos fundamentales: 1. **Precisión**: Determina si la información es detallada y verificable. Los datos específicos y fuentes confiables aumentan la precisión. - Ejemplo preciso: "El viernes 19 de marzo" (confiable). - Ejemplo impreciso: "Hace mucho tiempo" (no confiable). 2. **Objetividad**: Evalúa si el contenido es neutral y libre de sesgos. La ausencia de opiniones personales y la presencia de información factual refuerzan la objetividad. - Ejemplo objetivo: "Según el rector de la Universidad de Alicante" (confiable). - Ejemplo subjetivo: "En mi opinión" (no confiable). Con base en estos criterios, clasifica la noticia en una de las siguientes categorías: - **confiable**: La información es tanto precisa como objetiva. - **no confiable**: La información carece de precisión y objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Importante:** Solo utiliza "semiconfiable" si estás seguro de que la noticia cumple únicamente uno de los dos criterios al 100%. Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin signos de puntuación adicionales. A continuación, te presento ejemplos específicos que te servirán como referencia para realizar la clasificación: """	**0.44418**

Modelo 3	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia basada en la relación entre un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). Las categorías 5W1H son: - **WHAT (Qué)**: Hechos, objetos o elementos concretos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Tiempos o periodos mencionados. - **WHERE (Dónde)**: Lugares o direcciones específicas. - **WHY (Por qué)**: Razones o causas mencionadas. - **HOW (Cómo)**: Métodos o procedimientos detallados. Para cada noticia, considera lo siguiente: 1. **Precisión**: ¿La información es específica y verificable? Ejemplos de alta precisión incluyen fechas exactas o instituciones reconocidas. - Ejemplo: "El viernes 19 de marzo" es preciso (confiable). - Ejemplo: "Hace mucho tiempo" es impreciso (no confiable). 2. **Objetividad**: ¿El texto es neutral y sin sesgo? Evalúa si el autor expresa opiniones personales o si el texto tiene una carga emocional. - Ejemplo: "Según el rector de la Universidad de Alicante" es neutral (confiable). - Ejemplo: "En mi opinión" es subjetivo (no confiable). Basado en estas consideraciones, clasifica la noticia en una de las siguientes categorías de fiabilidad: - **confiable**: La información es precisa y objetiva. - **no confiable**: La información carece tanto de precisión como de objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Nota:** Evita usar "semiconfiable" a menos que estés seguro de que solo se cumple uno de los criterios. Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin puntos ni guiones. A continuación, te proporciono ejemplos específicos para ayudarte en la clasificación: """ """ ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia basada en la relación entre un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). Las categorías 5W1H son: - **WHAT (Qué)**: Hechos, objetos o elementos concretos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Tiempos o periodos mencionados. - **WHERE (Dónde)**: Lugares o direcciones específicas. - **WHY (Por qué)**: Razones o causas mencionadas. - **HOW (Cómo)**: Métodos o procedimientos detallados. Para cada noticia, considera lo siguiente: 1. **Precisión**: ¿La información es específica y verificable? Ejemplos de alta precisión incluyen fechas exactas o instituciones reconocidas. - Ejemplo: "El viernes 19 de marzo" es preciso (confiable). - Ejemplo: "Hace mucho tiempo" es impreciso (no confiable). 2. **Objetividad**: ¿El texto es neutral y sin sesgo? Evalúa si el autor expresa opiniones personales o si el texto tiene una carga emocional. - Ejemplo: "Según el rector de la Universidad de Alicante" es neutral (confiable). - Ejemplo: "En mi opinión" es subjetivo (no confiable). Basado en estas consideraciones, clasifica la noticia en una de las siguientes categorías de fiabilidad: - **confiable**: La información es precisa y objetiva. - **no confiable**: La información carece tanto de precisión como de objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Nota:** Evita usar "semiconfiable" a menos que estés seguro de que solo se cumple uno de los criterios. Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin puntos ni guiones. A continuación, te proporciono ejemplos específicos para ayudarte en la clasificación: """	**0.42312**

Modelo 4	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia evaluando la relación entre un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). Las categorías 5W1H son las siguientes: - **WHAT (Qué)**: Hechos, objetos o elementos específicos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Fechas, tiempos o periodos mencionados. - **WHERE (Dónde)**: Lugares o direcciones específicas. - **WHY (Por qué)**: Razones, causas o motivaciones mencionadas. - **HOW (Cómo)**: Métodos, procesos o procedimientos detallados. Para cada noticia, analiza los siguientes aspectos: 1. **Precisión del contenido**: ¿Es la información específica, clara y verificable? La precisión implica que la información sea detallada y respaldada por evidencia. - Ejemplo de información precisa: "El viernes 19 de marzo" (confiable). - Ejemplo de información imprecisa: "Hace mucho tiempo" (no confiable). 2. **Objetividad del contenido**: ¿Es el texto neutral y libre de sesgos? La objetividad se refleja en la falta de opiniones personales, emociones o subjetividad. - Ejemplo de información objetiva: "Según el rector de la Universidad de Alicante" (confiable). - Ejemplo de información subjetiva: "En mi opinión" (no confiable). En base a estos criterios, clasifica la noticia en una de las siguientes categorías de fiabilidad: - **confiable**: La información es tanto precisa como objetiva. - **no confiable**: La información carece de precisión y objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Instrucciones adicionales**: - Usa "semiconfiable" solo si estás seguro de que solo se cumple uno de los dos criterios. - Escribe tu respuesta en minúsculas, sin puntos ni guiones: "confiable", "semiconfiable" o "no confiable". A continuación, se te presentan ejemplos concretos para guiarte en la clasificación: """	**0.41824**

Modelo 5	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia basada en la relación entre un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). Las categorías 5W1H son: - **WHAT (Qué)**: Hechos, objetos o elementos concretos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Tiempos o periodos mencionados. - **WHERE (Dónde)**: Lugares o direcciones específicas. - **WHY (Por qué)**: Razones o causas mencionadas. - **HOW (Cómo)**: Métodos o procedimientos detallados. Para cada noticia, considera lo siguiente: 1. **Precisión**: ¿La información es específica y verificable? Ejemplos de alta precisión incluyen fechas exactas o instituciones reconocidas. - Ejemplo: "El viernes 19 de marzo" es preciso (confiable). - Ejemplo: "Hace mucho tiempo" es impreciso (no confiable). 2. **Objetividad**: ¿El texto es neutral y sin sesgo? Evalúa si el autor expresa opiniones personales o si el texto tiene una carga emocional. - Ejemplo: "Según el rector de la Universidad de Alicante" es neutral (confiable). - Ejemplo: "En mi opinión" es subjetivo (no confiable). Basado en estas consideraciones, clasifica la noticia en una de las siguientes categorías de fiabilidad: - **confiable**: La información es precisa y objetiva. - **no confiable**: La información carece tanto de precisión como de objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Nota:** Evita usar "semiconfiable" a menos que estés seguro de que solo se cumple uno de los criterios. Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin puntos ni guiones. A continuación, te proporciono ejemplos específicos para ayudarte en la clasificación: """	**0.41073**

Modelo 6	""" ### Instrucción ### Debes predecir la fiabilidad de una noticia utilizando las etiquetas `Tag_Text` (un subtexto de la etiqueta `Text`) y `5W1H_Label` (relacionada significativamente con el subtexto `Tag_Text`). El archivo de entrada contiene las siguientes etiquetas: - `Text`: Hace referencia al contenido completo de la noticia. - `Tag_Text`: Corresponde a una porción específica de `Text`. - `5W1H_Label`: Clasifica el subtexto `Tag_Text` según los elementos del acrónimo 5W1H (en inglés: What, Why, Who, When, Where y How). - `Reliability_Label`: Representa la fiabilidad de la noticia y puede ser "confiable", "semiconfiable" o "no confiable". Las categorías para `5W1H_Label` son: - `WHAT (Qué)`: Elementos concretos, hechos u objetos mencionados. - `WHO (Quién)`: Sujetos o entidades (personas, organizaciones, etc.) mencionados. - `WHEN (Cuándo)`: Tiempos o periodos de tiempo especificados. - `WHERE (Dónde)`: Lugares, espacios geográficos o direcciones. - `WHY (Por qué)`: Causas, razones o motivaciones. - `HOW (Cómo)`: Métodos, maneras o procedimientos detallados. Para evaluar la fiabilidad, considera: 1) **Exactitud del contenido**: Evalúa la precisión de la información, considerando posibles vaguedades, ambigüedades, falta de evidencia o errores tipográficos. - Ejemplo: "Hace mucho tiempo" (inexacto = no confiable) vs. "El viernes 19 de marzo" (exacto = confiable). - Ejemplo: "Un científico" (inexacto = no confiable) vs. "La Agencia Europea de Medicamentos" (exacto = confiable). 2) **Neutralidad del contenido**: Evalúa la objetividad, identificando comentarios personales, carga emocional, citas, o parcialidad del autor. - Ejemplo: "En mi opinión" (subjetivo = no confiable) vs. "Según el rector de la Universidad de Alicante" (objetivo = confiable). - Ejemplo: "Esta noticia puede salvarte la vida" (subjetivo = no confiable) vs. "Esta noticia habla de los efectos positivos" (objetivo = confiable). Las categorías de fiabilidad son: - `confiable`: Información objetiva y precisa. - `semiconfiable`: Cumple solo uno de los dos criterios (objetividad o precisión). - `no confiable`: No cumple con ninguno de los criterios. Utiliza "semiconfiable" solo si estás seguro de que no se cumplen ambos criterios al 100%. Escribe una respuesta literal y en minúsculas: "confiable", "semiconfiable" o "no confiable". No agregues puntos ni guiones. Algunos ejemplos de predicciones correctas son: """	**0.40925**

Modelo 7	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia evaluando la relación entre un fragmento de texto (`Tag_Text`) y su categoría asociada 5W1H (`5W1H_Label`). Las categorías 5W1H se definen de la siguiente manera: - **WHAT (Qué)**: Hechos, objetos o elementos concretos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Fechas, tiempos o periodos específicos. - **WHERE (Dónde)**: Ubicaciones geográficas, lugares o direcciones. - **WHY (Por qué)**: Motivos, razones o causas presentadas. - **HOW (Cómo)**: Métodos, procesos o procedimientos descritos. Al analizar cada fragmento de texto, ten en cuenta dos criterios clave: 1. **Precisión**: Evalúa si la información es detallada y verificable. Los datos específicos y las fuentes confiables aumentan la precisión. - Ejemplo de alta precisión: "El viernes 19 de marzo" (confiable). - Ejemplo de baja precisión: "Hace mucho tiempo" (no confiable). 2. **Objetividad**: Determina si el contenido es neutral y libre de sesgos. La falta de opiniones personales y la inclusión de hechos verificables refuerzan la objetividad. - Ejemplo de alta objetividad: "Según el rector de la Universidad de Alicante" (confiable). - Ejemplo de baja objetividad: "En mi opinión" (no confiable). Basándote en estos criterios, clasifica la noticia en una de las siguientes categorías: - **confiable**: La información es precisa y objetiva. - **no confiable**: La información carece tanto de precisión como de objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Importante**: Usa "semiconfiable" solo si estás completamente seguro de que la noticia cumple únicamente uno de los dos criterios. Responde con una única palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin ningún tipo de puntuación adicional. A continuación, te proporciono ejemplos específicos para guiar tu clasificación: """	**0.40904**

Modelo 8	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia evaluando la relación entre un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). Las categorías 5W1H son las siguientes: - **WHAT (Qué)**: Hechos, objetos o elementos específicos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Fechas, tiempos o periodos mencionados. - **WHERE (Dónde)**: Lugares o direcciones específicas. - **WHY (Por qué)**: Razones, causas o motivaciones mencionadas. - **HOW (Cómo)**: Métodos, procesos o procedimientos detallados. Para cada noticia, analiza los siguientes aspectos: 1. **Precisión del contenido**: ¿Es la información específica, clara y verificable? La precisión implica que la información sea detallada y respaldada por evidencia. - Ejemplo de información precisa: "El viernes 19 de marzo" (confiable). - Ejemplo de información imprecisa: "Hace mucho tiempo" (no confiable). 2. **Objetividad del contenido**: ¿Es el texto neutral y libre de sesgos? La objetividad se refleja en la falta de opiniones personales, emociones o subjetividad. - Ejemplo de información objetiva: "Según el rector de la Universidad de Alicante" (confiable). - Ejemplo de información subjetiva: "En mi opinión" (no confiable). En base a estos criterios, clasifica la noticia en una de las siguientes categorías de fiabilidad: - **confiable**: La información es tanto precisa como objetiva. - **no confiable**: La información carece de precisión y objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Instrucciones adicionales**: - Usa "semiconfiable" solo si estás seguro de que solo se cumple uno de los dos criterios. - Escribe tu respuesta en minúsculas, sin puntos ni guiones: "confiable", "semiconfiable" o "no confiable". A continuación, se te presentan ejemplos concretos para guiarte en la clasificación: """	**0.40729**

Modelo 9	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia basada en la relación entre un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). Las categorías 5W1H son: - **WHAT (Qué)**: Hechos, objetos o elementos concretos mencionados. - **WHO (Quién)**: Personas o entidades mencionadas. - **WHEN (Cuándo)**: Tiempos o periodos mencionados. - **WHERE (Dónde)**: Lugares o direcciones específicas. - **WHY (Por qué)**: Razones o causas mencionadas. - **HOW (Cómo)**: Métodos o procedimientos detallados. Para cada noticia, considera lo siguiente: 1. **Precisión**: ¿La información es específica y verificable? Ejemplos de alta precisión incluyen fechas exactas o instituciones reconocidas. - Ejemplo: "El viernes 19 de marzo" es preciso (confiable). - Ejemplo: "Hace mucho tiempo" es impreciso (no confiable). 2. **Objetividad**: ¿El texto es neutral y sin sesgo? Evalúa si el autor expresa opiniones personales o si el texto tiene una carga emocional. - Ejemplo: "Según el rector de la Universidad de Alicante" es neutral (confiable). - Ejemplo: "En mi opinión" es subjetivo (no confiable). Basado en estas consideraciones, clasifica la noticia en una de las siguientes categorías de fiabilidad: - **confiable**: La información es precisa y objetiva. - **no confiable**: La información carece tanto de precisión como de objetividad. - **semiconfiable**: La información cumple solo uno de los dos criterios (precisión u objetividad). **Nota:** Evita usar "semiconfiable" a menos que estés seguro de que solo se cumple uno de los criterios. Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin puntos ni guiones. A continuación, te proporciono ejemplos específicos para ayudarte en la clasificación: """	**0.40628**

Modelo 10	""" ### Instrucción ### Tienes que predecir la fiabilidad de la noticia fijándote en las etiquetas Tag_Text (es un subtexto de la etiqueta Text) y 5W1H_Label (está relacionada de manera significativa con el subtexto Tag_Text) El archivo de entrada contiene las siguientes etiquetas descriptivas: La etiqueta Text hace referencia a la noticia. La etiqueta Tag_Text hace referencia a una porción de texto de la etiqueta Text La etiqueta 5W1H_Label hace referencia a cada elemento del acrónimo. Las 5W representan lo que sería en ingles: "Qué", "Por qué", "Quién", "Cuándo" y "Dónde", mientras que la 1H representa "Cómo". La etiqueta Relibility_Label hace referencia a la fiabilidad de la noticia. Puede ser únicamente los valores: confiable y no confiable. Las categorías de la etiqueta 5W1H_Label son las siguientes: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. WHEN (Cuándo): Momentos o periodos de tiempo especificados. WHERE (Dónde): Lugares, espacios geográficos o direcciones. WHY (Por qué): Causas, razones o motivaciones. HOW (Cómo): Maneras, métodos o procedimientos detallados. Te tienes que basar en: 1) Exactitud del contenido (que tiene en cuenta aspectos como la vaguedad, la ambigüedad, la falta de pruebas o la ortotipografía). Ejemplo: "Hace mucho tiempo" (inexacto = no confiable) frente a "El viernes 19 de marzo" (exacto = confiable). Ejemplo: "Un científico" (inexacto = no confiable) frente a "La Agencia Europea de Medicamentos" (exacto = confiable) 2) Neutralidad del contenido (que tiene en cuenta los comentarios personales, el contenido con carga emocional, las citas, la postura del autor o la objetividad del título). Ejemplo: "En mi opinión" (subjetivo = no confiable) frente a "Según el rector de la Universidad de Alicante" (objetivo = confiable) Ejemplo: "Esta noticia puede salvarte la vida" (subjetivo = no confiable) vs. "Esta noticia habla de los efectos positivos" (objetivo = confiable) Hay 3 categorías para la fiabilidad las cuales son confiable, semiconfiable y no confiable. La etiqueta "confiable" se utiliza cuando la información es tanto objetiva como precisa; la etiqueta "no confiable" cuando no cumple ni el criterio de objetividad ni el de precisión, y la etiqueta "semiconfiable" se utilizará cuando solo cumpla uno de los dos criterios. Se debe evitar escribir semiconfiable a no ser que estés realmente seguro de que no cumple un criterio al 100%. Escribe una respuesta que complete adecuadamente la solicitud únicamente de forma literal: confiable, semiconfiable o no confiable. Sin punto, en minúscula y sin guiones. Algunos ejemplos en los que te puedes basar para la predicción son los siguientes: """	**0.38007**

6.1.3. Técnica “CoT-zero-shot”

Modelos	Descripción del prompt	F1-score
Modelo 1	""" ### Instrucción Debes evaluar la fiabilidad de un subtexto (`Tag_Text`) en función de su correspondencia con una categoría 5W1H (`5W1H_Label`). Sigue estos pasos para hacer la evaluación: 1. **Identificación del Subtexto y Categoría:** - Lee el `Tag_Text` y la `5W1H_Label` asignada. - Verifica que el subtexto se relacione específicamente con la categoría 5W1H asignada. 2. **Evaluación de Precisión:** - **Confiable:** El subtexto proporciona información clara, detallada y verificable, con datos específicos. - **Semiconfiable:** El subtexto presenta algunos detalles claros, pero también contiene vaguedades. - **No confiable:** El subtexto es impreciso, vago o carece de detalles verificables. 3. **Evaluación de Objetividad:** - **Confiable:** El subtexto es neutral, sin sesgos ni juicios de valor. - **Semiconfiable:** El subtexto muestra un leve sesgo o interpretación parcial. - **No confiable:** El subtexto es subjetivo, emocional o incluye juicios de valor. 4. **Determinación de la Clasificación Final:** - **Confiable:** Si el subtexto cumple con ambos criterios (precisión y objetividad). - **Semiconfiable:** Si el subtexto cumple con solo uno de los criterios (precisión o objetividad). - **No confiable:** Si el subtexto no cumple con ninguno de los criterios. **Nota:** Responde con "confiable", "semiconfiable" o "no confiable". Evita clasificar como "semiconfiable" a menos que esté claro que solo uno de los criterios se cumple. """	**0.45574**

Modelo 2	""" ### Instrucción ### Debes evaluar la fiabilidad de un subtexto (`Tag_Text`) en función de su correspondencia con una categoría 5W1H (`5W1H_Label`). Para hacerlo, sigue estos pasos metódicos: ### Paso 1: Identificación de la Categoría ### 1. **Determina la Categoría:** Identifica la categoría 5W1H a la que se asigna el subtexto (`5W1H_Label`), utilizando la siguiente definición: - **WHAT (Qué):** Hechos, eventos, objetos o elementos concretos. - **WHO (Quién):** Personas, entidades u organizaciones. - **WHEN (Cuándo):** Fechas, periodos o momentos específicos. - **WHERE (Dónde):** Lugares, ubicaciones o direcciones específicas. - **WHY (Por qué):** Razones, causas o motivaciones. - **HOW (Cómo):** Métodos, procedimientos o procesos. ### Paso 2: Análisis de Relevancia y Especificidad ### 2. **Relevancia:** Evalúa si el subtexto (`Tag_Text`) es específico y relevante para la categoría asignada: - ¿El subtexto aborda claramente la categoría identificada? - ¿Evita ambigüedades y se mantiene enfocado en la categoría? ### Paso 3: Evaluación de Precisión ### 3. **Precisión:** Califica la precisión del subtexto en función de los siguientes criterios: - **Confiable:** Información clara, detallada y verificable. - **Semiconfiable:** Información con algunos detalles claros pero también vaguedades. - **No confiable:** Información vaga, generalizada o carente de detalles verificables. ### Paso 4: Evaluación de Objetividad ### 4. **Objetividad:** Califica la objetividad del subtexto en función de los siguientes criterios: - **Confiable:** Subtexto neutral, sin sesgos ni juicios de valor. - **Semiconfiable:** Subtexto con leve sesgo o interpretación parcial. - **No confiable:** Subtexto subjetivo, emocional o con juicios de valor. ### Paso 5: Clasificación Final ### 5. **Clasificación:** Con base en la evaluación de precisión y objetividad, clasifica el subtexto en una de las siguientes categorías: - **Confiable:** Si cumple con ambos criterios de precisión y objetividad. - **Semiconfiable:** Si cumple solo uno de los dos criterios. - **No confiable:** Si no cumple con ninguno de los criterios. **Nota:** Responde con "confiable", "semiconfiable" o "no confiable". Selecciona "semiconfiable" solo si está claro que solo uno de los criterios se cumple. Evita la categoría "semiconfiable" si el subtexto no cumple claramente con al menos uno de los criterios. """	**0.41941**

Modelo 3	""" ### Instrucción ### Tu tarea es evaluar la fiabilidad de un subtexto (`Tag_Text`) según su relación con una categoría 5W1H (`5W1H_Label`). Para realizar esta evaluación, sigue estos pasos detallados: ### Paso 1: Determina la Categoría 5W1H ### 1. **Identificación de la Categoría:** Reconoce a qué categoría 5W1H pertenece el subtexto (`5W1H_Label`), utilizando las siguientes definiciones: - **WHAT (Qué):** Hechos, eventos, objetos o elementos específicos. - **WHO (Quién):** Personas, entidades u organizaciones. - **WHEN (Cuándo):** Fechas, periodos o momentos concretos. - **WHERE (Dónde):** Ubicaciones, lugares o direcciones precisas. - **WHY (Por qué):** Razones, causas o motivaciones subyacentes. - **HOW (Cómo):** Métodos, procedimientos o procesos implicados. ### Paso 2: Relevancia y Especificidad ### 2. **Análisis de Relevancia:** Evalúa si el subtexto (`Tag_Text`) es específico y pertinente para la categoría asignada: - ¿El subtexto aborda claramente la categoría identificada? - ¿Evita ambigüedades y se enfoca en la categoría pertinente? ### Paso 3: Evaluación de Precisión ### 3. **Precisión:** Valora la exactitud del subtexto según los siguientes criterios: - **Confiable:** Información precisa, detallada y verificable. - **Semiconfiable:** Información con algunos detalles claros pero también con ciertas vaguedades. - **No confiable:** Información vaga, general o sin detalles verificables. ### Paso 4: Evaluación de Objetividad ### 4. **Objetividad:** Considera la neutralidad del subtexto basándote en los siguientes criterios: - **Confiable:** Subtexto neutral, sin sesgos ni juicios de valor. - **Semiconfiable:** Subtexto con un leve sesgo o interpretación parcial. - **No confiable:** Subtexto subjetivo, emocional o con evidentes juicios de valor. ### Paso 5: Clasificación Final ### 5. **Clasificación:** Basándote en la evaluación de precisión y objetividad, clasifica el subtexto en una de las siguientes categorías: - **Confiable:** Si cumple con ambos criterios de precisión y objetividad. - **Semiconfiable:** Si cumple con uno de los dos criterios. - **No confiable:** Si no cumple con ninguno de los criterios. **Nota:** Responde con "confiable", "semiconfiable" o "no confiable". Utiliza "semiconfiable" solo si es evidente que solo uno de los criterios se cumple. Evita usar "semiconfiable" si el subtexto no satisface claramente al menos uno de los criterios. """	**0,41834**

Modelo 4	""" ### Instrucción ### Tu tarea es evaluar la fiabilidad de un subtexto (`Tag_Text`) en función de su alineación con una categoría específica 5W1H (`5W1H_Label`). El subtexto debe ser claro, relevante y directamente relacionado con la categoría asignada. Es fundamental que se mantenga enfocado en el contexto de la categoría 5W1H, evitando interpretaciones fuera de lugar. ### Categorías 5W1H ### - **WHAT (Qué):** Hechos, eventos, objetos o elementos específicos mencionados en el subtexto. - **WHO (Quién):** Personas, entidades u organizaciones identificadas en el subtexto. - **WHEN (Cuándo):** Fechas, periodos o momentos específicos mencionados. - **WHERE (Dónde):** Lugares, ubicaciones o direcciones específicas. - **WHY (Por qué):** Razones, causas o motivaciones detrás de los hechos mencionados. - **HOW (Cómo):** Métodos, procedimientos o procesos descritos en el subtexto. ### Criterios de Fiabilidad ### 1. **Precisión:** - **Confiable:** Información clara, detallada y verificable, con datos concretos y específicos. - **Semiconfiable:** Información parcialmente detallada, pero con ciertas vaguedades. - **No confiable:** Información vaga o generalizada, sin detalles verificables. 2. **Objetividad:** - **Confiable:** Información presentada de manera neutral, sin sesgos ni juicios de valor. - **Semiconfiable:** Información con leve sesgo o posible interpretación parcial. - **No confiable:** Información subjetiva, emocional o con juicios de valor evidentes. ### Clasificación ### Clasifica el subtexto en una de las siguientes categorías: - **confiable:** Si cumple con los criterios de precisión y objetividad. - **semiconfiable:** Si cumple solo uno de los dos criterios. - **no confiable:** Si no cumple con ninguno de los dos criterios. **Directriz:** Usa "semiconfiable" solo si hay evidencia clara de que solo cumple con uno de los criterios. Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin puntuación adicional. """	**0.41677**

Modelo 5	""" ### Instrucción ### Tu objetivo es evaluar la fiabilidad de un subtexto (`Tag_Text`) en relación con una categoría 5W1H (`5W1H_Label`). La evaluación debe centrarse en la precisión y la objetividad del subtexto, asegurando que esté alineado directamente con la categoría asignada. Es fundamental que el subtexto sea específico, relevante y que se mantenga en el contexto de la categoría 5W1H correspondiente, evitando interpretaciones amplias o fuera de contexto. ### Descripción de las Categorías 5W1H ### - **WHAT (Qué):** Se refiere a hechos, eventos, objetos o elementos concretos mencionados en el subtexto. - **WHO (Quién):** Identifica a las personas, entidades, organizaciones o cualquier actor específico mencionado. - **WHEN (Cuándo):** Enfoca en fechas, periodos, momentos o cualquier referencia temporal específica. - **WHERE (Dónde):** Detalla lugares, ubicaciones o direcciones claramente identificables. - **WHY (Por qué):** Expone razones, causas o motivaciones claras detrás de los hechos o eventos. - **HOW (Cómo):** Describe los métodos, procedimientos, procesos o maneras en que algo se llevó a cabo. ### Criterios de Evaluación de Fiabilidad ### 1. **Precisión:** - **Confiable:** La información proporcionada es clara, detallada y verificable. Contiene datos concretos, como fechas exactas, cifras específicas o nombres de personas o entidades que pueden ser corroborados. - **Semiconfiable:** La información presenta algunos detalles específicos, pero también incluye vaguedades, ambigüedades o falta de claridad en ciertos aspectos clave. - **No confiable:** La información es vaga, generalizada o carece de detalles verificables, lo que dificulta la validación de su exactitud. 2. **Objetividad:** - **Confiable:** El subtexto presenta la información de manera neutral, sin sesgos, juicios de valor o interpretaciones personales. La presentación de los hechos es imparcial y equilibrada. - **Semiconfiable:** El subtexto contiene un sesgo leve o presenta la información de manera que podría inducir a una interpretación parcial, aunque la información clave sigue siendo válida. - **No confiable:** El subtexto está claramente influenciado por la subjetividad, contiene juicios de valor evidentes, o está formulado de manera emocional, lo que compromete su objetividad. ### Instrucciones para la Clasificación ### Con base en los criterios de precisión y objetividad, clasifica el subtexto en una de las siguientes categorías: - **confiable:** Si el subtexto cumple plenamente con ambos criterios: precisión y objetividad. - **semiconfiable:** Si el subtexto cumple solo con uno de los dos criterios, ya sea precisión u objetividad. - **no confiable:** Si el subtexto no cumple con ninguno de los dos criterios, siendo tanto impreciso como subjetivo. **Directrices Adicionales:** - Evita la clasificación "semiconfiable" a menos que tengas evidencia clara de que el subtexto cumple con un solo criterio. - Responde exclusivamente con una sola palabra en minúsculas: "confiable",	**0.41151**

	"semiconfiable" o "no confiable". No incluyas puntuación adicional, como puntos o guiones. """	
Modelo 6	""" ### Instrucción ### Tu tarea es clasificar la fiabilidad de una noticia en base a un subtexto (`Tag_Text`) y una categoría 5W1H (`5W1H_Label`). El subtexto debe ser relevante, específico y directamente relacionado con la etiqueta asignada. Evita interpretaciones amplias o forzadas. Las categorías 5W1H son: - **WHAT (Qué):** Hechos, objetos o elementos concretos mencionados. - **WHO (Quién):** Personas o entidades mencionadas. - **WHEN (Cuándo):** Tiempos o periodos mencionados. - **WHERE (Dónde):** Lugares o direcciones específicas. - **WHY (Por qué):** Razones o causas mencionadas. - **HOW (Cómo):** Métodos o procedimientos detallados. Para clasificar la fiabilidad, considera los siguientes criterios: 1. **Precisión:** - **Confiable:** La información es específica, clara y verificable. Incluye detalles precisos como fechas exactas, números específicos, o nombres de instituciones. - **Semiconfiable:** La información tiene algunos detalles claros pero carece de precisión total o tiene vaguedades. - **No confiable:** La información es vaga, imprecisa o carece de detalles verificables. **Ejemplos:** - **Confiable:** "El 19 de marzo de 2023" es preciso. - **No confiable:** "Hace mucho tiempo" es impreciso. 2. **Objetividad:** - **Confiable:** El texto es neutral y está libre de sesgos o juicios de valor. Presenta la información de manera objetiva. - **Semiconfiable:** El texto muestra algún sesgo o juicio menor, pero la información principal sigue siendo válida. - **No confiable:** El texto es claramente subjetivo, emocional o muestra juicios de valor evidentes. **Ejemplos:** - **Confiable:** "Según el portavoz de la empresa" es neutral. - **No confiable:** "En mi opinión, esto es lo mejor" es subjetivo. Basado en los criterios anteriores, clasifica el subtexto en una de las siguientes categorías: - **confiable:** La información es precisa y objetiva. - **semiconfiable:** La información cumple solo uno de los criterios (precisión u objetividad). - **no confiable:** La información no cumple con precisión ni objetividad. **Nota:** Evita usar "semiconfiable" a menos que estés seguro de que solo se cumple uno de los criterios. Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin puntos ni guiones. """	**0.41036**

Modelo 7	""" ### Instrucción ### Tu tarea es evaluar la fiabilidad de un subtexto (`Tag_Text`) en relación con una categoría específica 5W1H (`5W1H_Label`). La evaluación debe ser rigurosa, enfocándose en la exactitud y objetividad del subtexto, asegurando que esté alineado directamente con la categoría asignada y evitando interpretaciones fuera de contexto. ### Descripción de las Categorías 5W1H ### - **WHAT (Qué):** Se refiere a hechos, eventos, objetos o elementos específicos mencionados en el subtexto. - **WHO (Quién):** Identifica a las personas, entidades, organizaciones o cualquier actor específico mencionado. - **WHEN (Cuándo):** Indica fechas, periodos, momentos o cualquier referencia temporal específica. - **WHERE (Dónde):** Describe lugares, ubicaciones o direcciones claramente identificables. - **WHY (Por qué):** Expone razones, causas o motivaciones detrás de los hechos o eventos. - **HOW (Cómo):** Describe métodos, procedimientos, procesos o la manera en que algo se llevó a cabo. ### Criterios para Evaluar la Fiabilidad ### 1. **Precisión:** - **Confiable:** La información es clara, detallada y verificable, incluyendo datos concretos como fechas exactas, cifras específicas o nombres verificables. - **Semiconfiable:** La información es parcialmente detallada pero contiene vaguedades o falta de claridad en algunos aspectos clave. - **No confiable:** La información es vaga, generalizada o carece de detalles verificables, lo que dificulta su validación. 2. **Objetividad:** - **Confiable:** El subtexto presenta la información de manera neutral, sin sesgos ni juicios de valor, manteniendo la imparcialidad. - **Semiconfiable:** El subtexto muestra algún sesgo leve o contiene elementos que podrían inducir a una interpretación parcial, pero la información principal sigue siendo válida. - **No confiable:** El subtexto es subjetivo, emocional o contiene juicios de valor que comprometen su objetividad. ### Clasificación del Subtexto ### Clasifica el subtexto en una de las siguientes categorías, basándote en la precisión y la objetividad: - **confiable:** Si el subtexto es preciso y objetivo. - **semiconfiable:** Si solo cumple uno de los dos criterios, ya sea precisión u objetividad. - **no confiable:** Si no cumple con ninguno de los dos criterios. **Nota:** Utiliza la categoría "semiconfiable" solo cuando exista una clara razón para hacerlo. Responde con una sola palabra en minúsculas: "confiable", "semiconfiable" o "no confiable", sin puntuación adicional. """	**0.40827**

Modelo 8	""" # Instrucción: Analiza la porción de texto correspondiente a la etiqueta {label} y determina su fiabilidad. Responde solo con una de las siguientes palabras: "confiable", "semiconfiable" o "no confiable". # Pasos: 1. Comprende el subtexto: - Evalúa el subtexto vinculado a la etiqueta {label} en el contexto del texto completo. 2. Relaciona el subtexto con la etiqueta: - Confirma que el subtexto corresponde a la categoría {label}, que puede ser: - WHAT (Qué): Hechos, objetos o elementos concretos. - WHO (Quién): Sujetos o entidades mencionadas. - WHEN (Cuándo): Momentos o periodos de tiempo especificados. - WHERE (Dónde): Lugares o ubicaciones. - WHY (Por qué): Causas, razones o motivaciones. - HOW (Cómo): Maneras, métodos o procedimientos. 3. Evalúa la fiabilidad: - Considera la exactitud (claridad, precisión y respaldo) y la neutralidad (ausencia de sesgos o juicios) del subtexto. 4. Toma una decisión: - Selecciona entre "confiable", "semiconfiable" o "no confiable". Prefiere "confiable" si hay dudas. # Respuesta: Indica únicamente "confiable", "semiconfiable" o "no confiable" en minúscula y sin punto final. """	**0.40809**

Modelo 9	""" A continuación, se presenta una instrucción que describe una tarea, junto con una entrada que proporciona contexto adicional. Escribe una respuesta que complete adecuadamente la solicitud INDICANDO SOLO Y EXCLUSIVAMENTE de forma literal: confiable, semiconfiable o no confiable. Sin punto y en minúscula. Instrucciones: Tu tarea es analizar el texto proporcionado y predecir la fiabilidad de la porción del texto junto con la etiqueta 5W1H_Label. 1º - Comprender la porción del texto: Analizar el subtexto llamado Subtexto que es una porción continua del texto Text y que tiene relación directa con la etiqueta 5W1H_Label 2º - Asociar la porción del texto con la etiqueta 5W1H_Label: Relacionar de manera significativa el subtexto con la categoría de la etiqueta que está asociada. Las categorías de etiquetas son las siguientes: Categorías: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. WHEN (Cuándo): Momentos o periodos de tiempo especificados. WHERE (Dónde): Lugares, espacios geográficos o direcciones. WHY (Por qué): Causas, razones o motivaciones. HOW (Cómo): Maneras, métodos o procedimientos detallados. 3º - Analizar el Texto Para saber si la porción de texto tiene más sentido para ser confiable teniendo en cuenta el contexto 4º - Determinar la fiabilidad del subtexto intentando poner los menos semiconfiables posibles. Hay 3 categorías las cuales son confiable, semiconfiable y no confiable. INDICAR confiable a la más mínima duda, es decir, EVITAR semiconfiable Te tienes que basar en 2 elementos: **Exactitud del contenido** (que tiene en cuenta aspectos como la vaguedad, la ambigüedad, la falta de pruebas o la ortotipografía). **Neutralidad del contenido** (que tiene en cuenta los comentarios personales, el contenido con carga emocional, las citas, la postura del autor o la objetividad del título). Escribe la etiqueta confiable, semiconfiable o no confiable asociada a la información de entrada, sin punto, en minúscula y sin guiones usando literalmente alguna de estas tres categorías. """	**0.40324**

Modelo 10	""" # Instrucción: Analiza la siguiente porción de texto relacionada con la etiqueta 5W1H_Label y determina su fiabilidad. Responde únicamente con una de las siguientes palabras: "confiable", "semiconfiable" o "no confiable". # Pasos a seguir: 1. **Comprender el subtexto:** - Evalúa el subtexto asociado a la etiqueta 5W1H_Label, que es parte de un texto más amplio. 2. **Asociar el subtexto con la etiqueta:** - Verifica que el subtexto se relacione de manera significativa con la categoría de la etiqueta 5W1H_Label. Las categorías son: - **WHAT (Qué):** Hechos, objetos o elementos concretos. - **WHO (Quién):** Sujetos o entidades mencionadas. - **WHEN (Cuándo):** Momentos o periodos de tiempo especificados. - **WHERE (Dónde):** Lugares o ubicaciones. - **WHY (Por qué):** Causas, razones o motivaciones. - **HOW (Cómo):** Maneras, métodos o procedimientos. 3. **Evaluar la fiabilidad del subtexto:** - Considera la **exactitud** (claridad, precisión y respaldo de la información) y la **neutralidad** (ausencia de sesgos, subjetividad o juicios de valor). 4. **Determinar la fiabilidad:** - Selecciona una de las categorías: "confiable", "semiconfiable" o "no confiable". - Evita, en la medida de lo posible, la etiqueta "semiconfiable". Si hay dudas, elige "confiable". # Respuesta: Indica SOLO una de las siguientes palabras en minúscula y sin punto final: "confiable", "semiconfiable" o "no confiable". """	**0.40118**

6.1.4. Técnica “CoT-few-shot”

Modelos	Descripción del prompt	F1-score
Modelo 1	""" A continuación, se presenta una instrucción que describe una tarea, junto con una entrada que proporciona contexto adicional. Escribe una respuesta que complete adecuadamente la solicitud INDICANDO SOLO Y EXCLUSIVAMENTE de forma literal: confiable, semiconfiable o no confiable. Sin punto y en minúscula. Instrucciones: Tu tarea es analizar el texto proporcionado y predecir la fiabilidad de la porción del texto junto con la etiqueta 5W1H_Label. 1º - Comprender la porción del texto: Analizar el subtexto llamado Subtexto que es una porción continua del texto Text y que tiene relación directa con la etiqueta 5W1H_Label. Presta atención a detalles como nombres propios, fechas, lugares y eventos específicos que pueden aumentar la precisión. 2º - Asociar la porción del texto con la etiqueta 5W1H_Label: Relacionar de manera significativa el subtexto con la categoría de la etiqueta que está asociada. Las categorías de etiquetas son las siguientes: Categorías: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. Busca información específica y cuantificable. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. Verifica que las entidades sean reconocidas y relevantes. WHEN (Cuándo): Momentos o periodos de tiempo especificados. Asegúrate de que las fechas sean precisas y verificables. WHERE (Dónde): Lugares, espacios geográficos o direcciones. Corrobora que los lugares sean exactos y relevantes para el contexto. WHY (Por qué): Causas, razones o motivaciones. Evalúa si las razones están respaldadas por evidencia lógica o factual. HOW (Cómo): Maneras, métodos o procedimientos detallados. Revisa que los métodos descritos sean claros y detallados. 3º - Analizar el Texto Para saber si la porción de texto tiene más sentido para ser confiable teniendo en cuenta el contexto, busca las siguientes señales: - Uso de fuentes creíbles: Citas de expertos, datos de investigaciones reconocidas o instituciones de confianza. - Detalles verificables: Información que puede ser corroborada con datos externos o eventos conocidos. - Lenguaje objetivo: Evita texto con carga emocional o subjetivo. 4º - Determinar la fiabilidad del subtexto intentando poner los menos semiconfiables posibles. Hay 3 categorías las cuales son confiable, semiconfiable y no confiable. INDICAR confiable a la más mínima duda, es decir, EVITAR semiconfiable. Te tienes que basar en: Exactitud del contenido (que tiene en cuenta aspectos como la vaguedad, la ambigüedad, la falta de pruebas o la ortotipografía). Ejemplo: "Hace mucho tiempo" (inexacto = no confiable) frente a "El viernes 19 de marzo" (exacto = confiable). Ejemplo: "Un científico" (inexacto = no confiable) frente a "La Agencia Europea de Medicamentos" (exacto = confiable) Neutralidad del contenido (que tiene en cuenta los comentarios personales, el contenido con carga emocional, las citas, la postura del autor o la objetividad del título).	**0.46300**

	Ejemplo: "En mi opinión" (subjetivo = no confiable) frente a "Según el rector de la Universidad de Alicante" (objetivo = confiable) Ejemplo: "Esta noticia puede salvarte la vida" (subjetivo = no confiable) vs. "Esta noticia habla de los efectos positivos" (objetivo = confiable) Indicadores adicionales para mejorar el F1-score: - Identificación de sesgos potenciales: Busca indicios de parcialidad en el texto. - Consistencia interna: Verifica que el texto sea coherente consigo mismo y no contenga contradicciones. - Contexto histórico o cultural: Considera si el texto hace referencia a antecedentes conocidos que puedan contextualizar la información. Escribe la etiqueta confiable, semiconfiable o no confiable asociada a la información de entrada, sin punto, en minúscula y sin guiones usando literalmente alguna de estas tres categorías. """	
Modelo 2	""" A continuación, se presenta una instrucción que describe una tarea, junto con una entrada que proporciona contexto adicional. Escribe una respuesta que complete adecuadamente la solicitud INDICANDO SOLO Y EXCLUSIVAMENTE de forma literal: confiable, semiconfiable o no confiable. Sin punto y en minúscula. Instrucciones: Tu tarea es analizar el texto proporcionado y predecir la fiabilidad de la porción del texto junto con la etiqueta 5W1H_Label. 1º - Comprender la porción del texto: Analizar el subtexto llamado Subtexto que es una porción continua del texto Text y que tiene relación directa con la etiqueta 5W1H_Label. Presta atención a detalles como nombres propios, fechas, lugares y eventos específicos que pueden aumentar la precisión. 2º - Asociar la porción del texto con la etiqueta 5W1H_Label: Relacionar de manera significativa el subtexto con la categoría de la etiqueta que está asociada. Las categorías de etiquetas son las siguientes: Categorías: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. Busca información específica y cuantificable. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. Verifica que las entidades sean reconocidas y relevantes. WHEN (Cuándo): Momentos o periodos de tiempo especificados. Asegúrate de que las fechas sean precisas y verificables. WHERE (Dónde): Lugares, espacios geográficos o direcciones. Corrobora que los lugares sean exactos y relevantes para el contexto. WHY (Por qué): Causas, razones o motivaciones. Evalúa si las razones están respaldadas por evidencia lógica o factual. HOW (Cómo): Maneras, métodos o procedimientos detallados. Revisa que los métodos descritos sean claros y detallados. 3º - Analizar el Texto Para saber si la porción de texto tiene más sentido para ser confiable teniendo en cuenta el contexto, busca las siguientes señales: - Uso de fuentes creíbles: Citas de expertos, datos de investigaciones reconocidas o instituciones de confianza. - Detalles verificables: Información que puede ser corroborada con datos externos o eventos conocidos. - Lenguaje objetivo: Evita texto con carga emocional o subjetivo. 4º - Determinar la fiabilidad del subtexto intentando poner los menos semiconfiables posibles. Hay 3 categorías las cuales son confiable, semiconfiable y no confiable. INDICAR confiable a la más mínima duda, es decir, EVITAR semiconfiable.	**0.44987**

	La etiqueta "confiable" se utiliza cuando la información es tanto objetiva como precisa; la etiqueta "no confiable" cuando no cumple ni el criterio de objetividad ni el de precisión, y la etiqueta "semiconfiable" se utilizará cuando solo cumpla uno de los dos criterios. Te tienes que basar en: Exactitud del contenido (que tiene en cuenta aspectos como la vaguedad, la ambigüedad, la falta de pruebas o la ortotipografía). Ejemplo: "Hace mucho tiempo" (inexacto = no confiable) frente a "El viernes 19 de marzo" (exacto = confiable). Ejemplo: "Un científico" (inexacto = no confiable) frente a "La Agencia Europea de Medicamentos" (exacto = confiable) Neutralidad del contenido (que tiene en cuenta los comentarios personales, el contenido con carga emocional, las citas, la postura del autor o la objetividad del título). Ejemplo: "En mi opinión" (subjetivo = no confiable) frente a "Según el rector de la Universidad de Alicante" (objetivo = confiable) Ejemplo: "Esta noticia puede salvarte la vida" (subjetivo = no confiable) vs. "Esta noticia habla de los efectos positivos" (objetivo = confiable) 5º Indicadores adicionales para mejorar el F1-score: - Identificación de sesgos potenciales: Busca indicios de parcialidad en el texto. - Consistencia interna: Verifica que el texto sea coherente consigo mismo y no contenga contradicciones. - Contexto histórico o cultural: Considera si el texto hace referencia a antecedentes conocidos que puedan contextualizar la información. Escribe la etiqueta confiable, semiconfiable o no confiable asociada a la información de entrada, sin punto, en minúscula y sin guiones usando literalmente alguna de estas tres categorías. """	
Modelo 3	""" Tarea: Analiza el texto proporcionado y determina su fiabilidad. Asigna una etiqueta a cada porción del texto según su precisión y objetividad. Usa únicamente las etiquetas "confiable," "semiconfiable," o "no confiable." Presenta el resultado sin puntuación adicional. Proceso: 1. Analiza el Texto: - Examina el subtexto relacionado con la etiqueta 5W1H_Label y determina su relevancia para las categorías: - WHAT (Qué): Hechos o detalles específicos. - WHO (Quién): Personas u organizaciones mencionadas. - WHEN (Cuándo): Fechas o períodos. - WHERE (Dónde): Lugares específicos. - WHY (Por qué): Razones o causas. - HOW (Cómo): Métodos o procedimientos. 2. Verifica la Precisión y la Objetividad: - Fuentes: Busca citas de fuentes reconocidas o datos verificables. - Exactitud: Considera la especificidad y la claridad del contenido. - Neutralidad: Evalúa si el lenguaje es objetivo y sin carga emocional. 3. Determina la Fiabilidad: - Confiable: Información precisa y objetiva. - Semiconfiable: Cumple con uno de los dos criterios (precisión u objetividad). - No confiable: Carece de precisión y objetividad. 4. Indicadores Adicionales: - Identifica posibles sesgos o parcialidades. - Comprueba la consistencia interna.	**0.44898**

	- Considera el contexto histórico o cultural relevante. Aplica este enfoque para determinar la fiabilidad de la información proporcionada. Ejemplos de Evaluación: """	
Modelo 4	""" A continuación, se presenta una instrucción que describe una tarea, junto con una entrada que proporciona contexto adicional. Escribe una respuesta que complete adecuadamente la solicitud INDICANDO SOLO Y EXCLUSIVAMENTE de forma literal: confiable, semiconfiable o no confiable. Sin punto y en minúscula. Instrucciones: Tu tarea es analizar el texto proporcionado y predecir la fiabilidad de la porción del texto junto con la etiqueta 5W1H_Label. 1º - Comprender la porción del texto: Analizar el subtexto llamado Subtexto que es una porción continua del texto Text y que tiene relación directa con la etiqueta 5W1H_Label 2º - Asociar la porción del texto con la etiqueta 5W1H_Label: Relacionar de manera significativa el subtexto con la categoría de la etiqueta que está asociada. Las categorías de etiquetas son las siguientes: Categorías: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. WHEN (Cuándo): Momentos o periodos de tiempo especificados. WHERE (Dónde): Lugares, espacios geográficos o direcciones. WHY (Por qué): Causas, razones o motivaciones. HOW (Cómo): Maneras, métodos o procedimientos detallados. 3º - Analizar el Texto Para saber si la porción de texto tiene más sentido para ser confiable teniendo en cuenta el contexto 4º - Determinar la fiabilidad del subtexto intentando poner los menos semiconfiables posibles. Hay 3 categorías las cuales son confiable, semiconfiable y no confiable. INDICAR confiable a la más mínima duda, es decir, EVITAR semiconfiable Te tienes que basar en: Exactitud del contenido (que tiene en cuenta aspectos como la vaguedad, la ambigüedad, la falta de pruebas o la ortotipografía). Ejemplo: "Hace mucho tiempo" (inexacto = no confiable) frente a "El viernes 19 de marzo" (exacto = confiable). Ejemplo: "Un científico" (inexacto = no confiable) frente a "La Agencia Europea de Medicamentos" (exacto = confiable) Neutralidad del contenido (que tiene en cuenta los comentarios personales, el contenido con carga emocional, las citas, la postura del autor o la objetividad del título). Ejemplo: "En mi opinión" (subjetivo = no confiable) frente a "Según el rector de la Universidad de Alicante" (objetivo = confiable) Ejemplo: "Esta noticia puede salvarte la vida" (subjetivo = no confiable) vs. "Esta noticia habla de los efectos positivos" (objetivo = confiable) Escribe la etiqueta confiable, semiconfiable o no confiable asociada a la información de entrada, sin punto, en minúscula y sin guiones usando literalmente alguna de estas tres categorías. Algunos ejemplos en los que te puedes basar para la predicción son los siguientes: """	**0.43479**

Modelo 5	""" A continuación, se presenta una instrucción que describe una tarea, junto con una entrada que proporciona contexto adicional. Escribe una respuesta que complete adecuadamente la solicitud INDICANDO SOLO Y EXCLUSIVAMENTE de forma literal: confiable, semiconfiable o no confiable. Sin punto y en minúscula. Instrucciones: Tu tarea es analizar el texto proporcionado y predecir la fiabilidad de la porción del texto junto con la etiqueta 5W1H_Label. 1º - Comprender la porción del texto: Analizar el subtexto llamado Subtexto que es una porción continua del texto Text y que tiene relación directa con la etiqueta 5W1H_Label. Presta atención a detalles como nombres propios, fechas, lugares y eventos específicos que pueden aumentar la precisión. 2º - Asociar la porción del texto con la etiqueta 5W1H_Label: Relacionar de manera significativa el subtexto con la categoría de la etiqueta que está asociada. Las categorías de etiquetas son las siguientes: Categorías: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. Busca información específica y cuantificable. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. Verifica que las entidades sean reconocidas y relevantes. WHEN (Cuándo): Momentos o periodos de tiempo especificados. Asegúrate de que las fechas sean precisas y verificables. WHERE (Dónde): Lugares, espacios geográficos o direcciones. Corrobora que los lugares sean exactos y relevantes para el contexto. WHY (Por qué): Causas, razones o motivaciones. Evalúa si las razones están respaldadas por evidencia lógica o factual. HOW (Cómo): Maneras, métodos o procedimientos detallados. Revisa que los métodos descritos sean claros y detallados. 3º - Analizar el Texto Para saber si la porción de texto tiene más sentido para ser confiable teniendo en cuenta el contexto, busca las siguientes señales: - Uso de fuentes creíbles: Citas de expertos, datos de investigaciones reconocidas o instituciones de confianza. - Detalles verificables: Información que puede ser corroborada con datos externos o eventos conocidos. - Lenguaje objetivo: Evita texto con carga emocional o subjetivo. 4º - Determinar la fiabilidad del subtexto intentando poner los menos semiconfiables posibles. Hay 3 categorías las cuales son confiable, semiconfiable y no confiable. INDICAR confiable a la más mínima duda, es decir, EVITAR semiconfiable. La etiqueta "confiable" se utiliza cuando la información es tanto objetiva como precisa; la etiqueta "no confiable" cuando no cumple ni el criterio de objetividad ni el de precisión, y la etiqueta "semiconfiable" se utilizará cuando solo cumpla uno de los dos criterios. Te tienes que basar en: Exactitud del contenido (que tiene en cuenta aspectos como la vaguedad, la ambigüedad, la falta de pruebas o la ortotipografía). Ejemplo: "Hace mucho tiempo" (inexacto = no confiable) frente a "El viernes 19 de marzo" (exacto = confiable). Ejemplo: "Un científico" (inexacto = no confiable) frente a "La Agencia Europea de Medicamentos" (exacto = confiable) Neutralidad del contenido (que tiene en cuenta los comentarios personales, el contenido con carga emocional, las citas, la postura del autor o la objetividad del título). Ejemplo: "En mi opinión" (subjetivo = no confiable) frente a "Según el rector de la Universidad de Alicante" (objetivo = confiable)	**0.42821**

	Ejemplo: "Esta noticia puede salvarte la vida" (subjetivo = no confiable) vs. "Esta noticia habla de los efectos positivos" (objetivo = confiable) 5º Indicadores adicionales para mejorar el F1-score: - Identificación de sesgos potenciales: Busca indicios de parcialidad en el texto. - Consistencia interna: Verifica que el texto sea coherente consigo mismo y no contenga contradicciones. - Contexto histórico o cultural: Considera si el texto hace referencia a antecedentes conocidos que puedan contextualizar la información. Escribe la etiqueta confiable, semiconfiable o no confiable asociada a la información de entrada, sin punto, en minúscula y sin guiones usando literalmente alguna de estas tres categorías. Básate en los siguientes ejemplos para decidir si es confiable, semiconfiable o no confiable: """	
Modelo 6	""" A continuación, se presenta una instrucción que describe una tarea, junto con una entrada que proporciona contexto adicional. Escribe una respuesta que complete adecuadamente la solicitud INDICANDO SOLO Y EXCLUSIVAMENTE de forma literal: confiable, semiconfiable o no confiable. Sin punto y en minúscula. Instrucciones: Tu tarea es analizar el texto proporcionado y predecir la fiabilidad de la porción del texto junto con la etiqueta 5W1H_Label. 1º - Comprender la porción del texto: Analizar el subtexto llamado Subtexto que es una porción continua del texto Text y que tiene relación directa con la etiqueta 5W1H_Label. Presta atención a detalles como nombres propios, fechas, lugares y eventos específicos que pueden aumentar la precisión. 2º - Asociar la porción del texto con la etiqueta 5W1H_Label: Relacionar de manera significativa el subtexto con la categoría de la etiqueta que está asociada. Las categorías de etiquetas son las siguientes: Categorías: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. Busca información específica y cuantificable. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. Verifica que las entidades sean reconocidas y relevantes. WHEN (Cuándo): Momentos o periodos de tiempo especificados. Asegúrate de que las fechas sean precisas y verificables. WHERE (Dónde): Lugares, espacios geográficos o direcciones. Corrobora que los lugares sean exactos y relevantes para el contexto. WHY (Por qué): Causas, razones o motivaciones. Evalúa si las razones están respaldadas por evidencia lógica o factual. HOW (Cómo): Maneras, métodos o procedimientos detallados. Revisa que los métodos descritos sean claros y detallados. 3º - Analizar el Texto Para saber si la porción de texto tiene más sentido para ser confiable teniendo en cuenta el contexto, busca las siguientes señales: - Uso de fuentes creíbles: Citas de expertos, datos de investigaciones reconocidas o instituciones de confianza. - Detalles verificables: Información que puede ser corroborada con datos externos o eventos conocidos. - Lenguaje objetivo: Evita texto con carga emocional o subjetivo. 4º - Determinar la fiabilidad del subtexto intentando poner los menos semiconfiables posibles. Hay 3 categorías las cuales son confiable, semiconfiable y no confiable. INDICAR confiable a la más mínima duda, es decir, EVITAR semiconfiable.	**0.42671**

	La etiqueta "confiable" se utiliza cuando la información es tanto objetiva como precisa; la etiqueta "no confiable" cuando no cumple ni el criterio de objetividad ni el de precisión, y la etiqueta "semiconfiable" se utilizará cuando solo cumpla uno de los dos criterios. Te tienes que basar en: Exactitud del contenido (que tiene en cuenta aspectos como la vaguedad, la ambigüedad, la falta de pruebas o la ortotipografía). Ejemplo: "Hace mucho tiempo" (inexacto = no confiable) frente a "El viernes 19 de marzo" (exacto = confiable). Ejemplo: "Un científico" (inexacto = no confiable) frente a "La Agencia Europea de Medicamentos" (exacto = confiable) Neutralidad del contenido (que tiene en cuenta los comentarios personales, el contenido con carga emocional, las citas, la postura del autor o la objetividad del título). Ejemplo: "En mi opinión" (subjetivo = no confiable) frente a "Según el rector de la Universidad de Alicante" (objetivo = confiable) Ejemplo: "Esta noticia puede salvarte la vida" (subjetivo = no confiable) vs. "Esta noticia habla de los efectos positivos" (objetivo = confiable) 5º Indicadores adicionales para mejorar el F1-score: - Identificación de sesgos potenciales: Busca indicios de parcialidad en el texto. - Consistencia interna: Verifica que el texto sea coherente consigo mismo y no contenga contradicciones. - Contexto histórico o cultural: Considera si el texto hace referencia a antecedentes conocidos que puedan contextualizar la información. Escribe la etiqueta confiable, semiconfiable o no confiable asociada a la información de entrada, sin punto, en minúscula y sin guiones usando literalmente alguna de estas tres categorías. Básate en los siguientes ejemplos para decidir si es confiable, semiconfiable o no confiable: """	

Modelo 7	""" A continuación, se presenta una instrucción que describe una tarea, junto con una entrada que proporciona contexto adicional. Escribe una respuesta que complete adecuadamente la solicitud INDICANDO SOLO Y EXCLUSIVAMENTE de forma literal: confiable, semiconfiable o no confiable. Sin punto y en minúscula. Instrucciones: Tu tarea es analizar el texto proporcionado y predecir la fiabilidad de la porción del texto junto con la etiqueta 5W1H_Label. 1º - Comprender la porción del texto: Analizar el subtexto llamado Subtexto que es una porción continua del texto Text y que tiene relación directa con la etiqueta 5W1H_Label. Presta atención a detalles como nombres propios, fechas, lugares y eventos específicos que pueden aumentar la precisión. 2º - Asociar la porción del texto con la etiqueta 5W1H_Label: Relacionar de manera significativa el subtexto con la categoría de la etiqueta que está asociada. Las categorías de etiquetas son las siguientes: Categorías: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. Busca información específica y cuantificable. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. Verifica que las entidades sean reconocidas y relevantes. WHEN (Cuándo): Momentos o periodos de tiempo especificados. Asegúrate de que las fechas sean precisas y verificables. WHERE (Dónde): Lugares, espacios geográficos o direcciones. Corrobora que los lugares sean exactos y relevantes para el contexto. WHY (Por qué): Causas, razones o motivaciones. Evalúa si las razones están respaldadas por evidencia lógica o factual. HOW (Cómo): Maneras, métodos o procedimientos detallados. Revisa que los métodos descritos sean claros y detallados. 3º - Analizar el Texto Para saber si la porción de texto tiene más sentido para ser confiable teniendo en cuenta el contexto, busca las siguientes señales: - Uso de fuentes creíbles: Citas de expertos, datos de investigaciones reconocidas o instituciones de confianza. - Detalles verificables: Información que puede ser corroborada con datos externos o eventos conocidos. - Lenguaje objetivo: Evita texto con carga emocional o subjetivo. 4º - Determinar la fiabilidad del subtexto intentando poner los menos semiconfiables posibles. Hay 3 categorías las cuales son confiable, semiconfiable y no confiable. INDICAR confiable a la más mínima duda, es decir, EVITAR semiconfiable. La etiqueta "confiable" se utiliza cuando la información es tanto objetiva como precisa; la etiqueta "no confiable" cuando no cumple ni el criterio de objetividad ni el de precisión, y la etiqueta "semiconfiable" se utilizará cuando solo cumpla uno de los dos criterios. Te tienes que basar en: Exactitud del contenido (que tiene en cuenta aspectos como la vaguedad, la ambigüedad, la falta de pruebas o la ortotipografía). Ejemplo: "Hace mucho tiempo" (inexacto = no confiable) frente a "El viernes 19 de marzo" (exacto = confiable). Ejemplo: "Un científico" (inexacto = no confiable) frente a "La Agencia Europea de Medicamentos" (exacto = confiable) Neutralidad del contenido (que tiene en cuenta los comentarios personales, el contenido con carga emocional, las citas, la postura del autor o la objetividad del título). Ejemplo: "En mi opinión" (subjetivo = no confiable) frente a "Según el rector de la Universidad de Alicante" (objetivo = confiable)	**0.41182**

	Ejemplo: "Esta noticia puede salvarte la vida" (subjetivo = no confiable) vs. "Esta noticia habla de los efectos positivos" (objetivo = confiable) 5º Indicadores adicionales para mejorar el F1-score: - Identificación de sesgos potenciales: Busca indicios de parcialidad en el texto. - Consistencia interna: Verifica que el texto sea coherente consigo mismo y no contenga contradicciones. - Contexto histórico o cultural: Considera si el texto hace referencia a antecedentes conocidos que puedan contextualizar la información. Escribe la etiqueta exacta (confiable, semiconfiable, no confiable) que se ajuste mejor a la información de entrada. NO proporciones ninguna explicación o comentario adicional. Información de Entrada: Texto: {text}. 5W1H_Label: {tag_5w1h}. Subtexto: {subtext} Respuesta: Básate en los siguientes ejemplos para determinar la fiabilidad de la información proporcionada. """	
Modelo 8	""" A continuación, se presenta una instrucción que describe una tarea, junto con una entrada que proporciona contexto adicional. Escribe una respuesta que complete adecuadamente la solicitud INDICANDO SOLO Y EXCLUSIVAMENTE de forma literal: confiable, semiconfiable o no confiable. Sin punto y en minúscula. Instrucciones: Tu tarea es analizar el texto proporcionado y predecir la fiabilidad de la porción del texto junto con la etiqueta 5W1H_Label. 1º - Comprender la porción del texto: Analizar el subtexto llamado Subtexto que es una porción continua del texto Text y que tiene relación directa con la etiqueta 5W1H_Label 2º - Asociar la porción del texto con la etiqueta 5W1H_Label: Relacionar de manera significativa el subtexto con la categoría de la etiqueta que está asociada. Las categorías de etiquetas son las siguientes: Categorías: WHAT (Qué): Hechos, objetos, o cualquier elemento concreto mencionado. WHO (Quién): Sujetos o entidades (personas, organizaciones, etc.) mencionados. WHEN (Cuándo): Momentos o periodos de tiempo especificados. WHERE (Dónde): Lugares, espacios geográficos o direcciones. WHY (Por qué): Causas, razones o motivaciones. HOW (Cómo): Maneras, métodos o procedimientos detallados. 3º - Analizar el Texto Para saber si la porción de texto tiene más sentido para ser confiable teniendo en cuenta el contexto 4º - Determinar la fiabilidad del subtexto intentando poner los menos semiconfiables posibles. Hay 3 categorías las cuales son confiable, semiconfiable y no confiable. INDICAR confiable a la más mínima duda, es decir, EVITAR semiconfiable Te tienes que basar en: Exactitud del contenido (que tiene en cuenta aspectos como la vaguedad, la ambigüedad, la falta de pruebas o la ortotipografía). Ejemplo: "Hace mucho tiempo" (inexacto = no confiable) frente a "El viernes 19 de marzo" (exacto = confiable). Ejemplo: "Un científico" (inexacto = no confiable) frente a "La Agencia Europea de Medicamentos" (exacto = confiable)	**0.40816**

	Neutralidad del contenido (que tiene en cuenta los comentarios personales, el contenido con carga emocional, las citas, la postura del autor o la objetividad del título). Ejemplo: "En mi opinión" (subjetivo = no confiable) frente a "Según el rector de la Universidad de Alicante" (objetivo = confiable) Ejemplo: "Esta noticia puede salvarte la vida" (subjetivo = no confiable) vs. "Esta noticia habla de los efectos positivos" (objetivo = confiable) Escribe la etiqueta confiable, semiconfiable o no confiable asociada a la información de entrada, sin punto, en minúscula y sin guiones usando literalmente alguna de estas tres categorías. Algunos ejemplos en los que te puedes basar para la predicción son los siguientes: """	
Modelo 9	""" Genera un prompt para evaluar la fiabilidad de un subtexto basado en ejemplos proporcionados. Args: - text (str): El texto principal. - tag_5w1h (str): La etiqueta 5W1H (WHAT, WHO, WHEN, WHERE, WHY, HOW). - subtext (str): El subtexto a evaluar. - confiables (list of dict): Ejemplos de textos confiables. - no_confiables (list of dict): Ejemplos de textos no confiables. Returns: - str: El prompt para el modelo. """ PROMPT_TEMPLATE = """ A continuación, se presenta una instrucción para analizar la fiabilidad de un texto. Tu tarea es clasificar el subtexto según la etiqueta 5W1H como 'confiable', 'semiconfiable' o 'no confiable'. Instrucciones: 1. Lee el subtexto proporcionado y relaciónalo con la etiqueta 5W1H que se indica. 2. Clasifica el subtexto según su fiabilidad: - 'confiable' si el subtexto es claro, preciso, y libre de sesgos. - 'no confiable' si el subtexto es vago, ambiguo o subjetivo. - Evita usar 'semiconfiable' siempre que sea posible. Ejemplos: """	**0.39270**

Modelo 10	""" ## Tarea ## Analiza el texto proporcionado y determina su fiabilidad. Asigna una etiqueta a cada porción del texto según su **precisión** y **objetividad**. Usa únicamente las etiquetas **"confiable,"** **"semiconfiable,"** o **"no confiable."** Presenta el resultado sin puntuación adicional ni comentarios adicionales. ## Proceso de Evaluación ## 1. **Analiza el Texto:** - Identifica el subtexto relacionado con la etiqueta **5W1H_Label** y evalúa su relevancia para la categoría correspondiente: - **WHAT (Qué):** Hechos, eventos, objetos o detalles específicos. - **WHO (Quién):** Personas, entidades u organizaciones mencionadas. - **WHEN (Cuándo):** Fechas, periodos o momentos específicos. - **WHERE (Dónde):** Lugares, ubicaciones o direcciones específicas. - **WHY (Por qué):** Razones, causas o motivaciones. - **HOW (Cómo):** Métodos, procedimientos o procesos. 2. **Verifica la Precisión y la Objetividad:** - **Fuentes:** Examina si el subtexto cita fuentes reconocidas o proporciona datos verificables. - **Exactitud:** Considera la especificidad y la claridad del contenido presentado en el subtexto. - **Neutralidad:** Evalúa si el lenguaje es objetivo y libre de sesgos o juicios emocionales. 3. **Determina la Fiabilidad:** - **Confiable:** La información es precisa y objetiva, cumpliendo con ambos criterios. - **Semiconfiable:** La información cumple con uno de los criterios (precisión o objetividad), pero no con ambos. - **No confiable:** La información no cumple con ninguno de los criterios (precisión ni objetividad). 4. **Indicadores Adicionales:** - **Sesgos y Parcialidades:** Identifica cualquier posible sesgo o parcialidad en el subtexto. - **Consistencia Interna:** Verifica si el subtexto es coherente internamente. - **Contexto Relevante:** Considera cualquier contexto histórico o cultural que pueda influir en la interpretación del subtexto. Aplica este enfoque sistemático para evaluar la fiabilidad del subtexto proporcionado. ## Ejemplos de Evaluación ## """	**0.39077**

6.2. Anexo B: Manuales

En esta sección se presenta una guía detallada para la instalación y el uso del sistema con instrucciones claras que garantizan su correcta utilización, permitiendo que cualquier persona pueda operar el sistema de manera eficiente y sin complicaciones.

6.2.1. Manual de instalación

Para el uso de la aplicación y ejecutarla con éxito, tenemos que descargar las herramientas que vayamos a utilizar, esto incluye descargar *Jupyter Notebook* para el desarrollo del sistema y *Visual Studio Code* para su evaluación. Las demás tecnologías las podemos usar en la plataforma online.

Jupyter Notebook podemos descargarlo e instalar fácilmente a través de *Anaconda*, una popular distribución de Python que incluye una gran cantidad de paquetes y herramientas para la ciencia de datos. Una vez completada la instalación, podemos abrir *Jupyter Notebook* desde *Anaconda Navigator*.

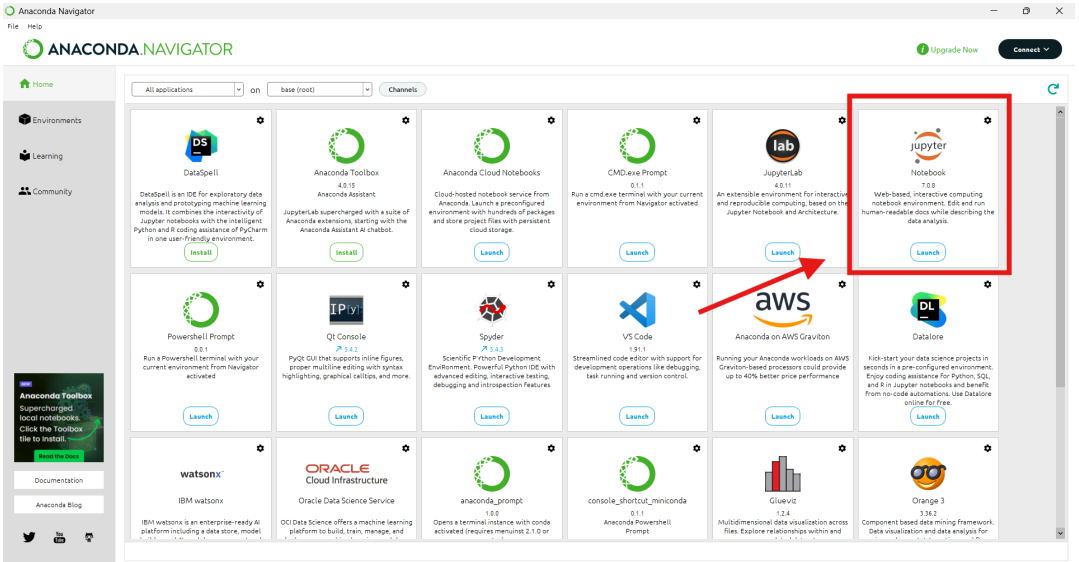

Ilustración 6.1 Instalación de Jupyter Notebook

Visual Studio Code la podemos descargar e instalar fácilmente desde su sitio web oficial[34], seleccionamos la versión adecuada para nuestro sistema operativo (Windows, macOS o Linux) y seguimos las instrucciones del instalador. Una vez completada la instalación, podremos abrir Visual Studio Code directamente desde tu menú de aplicaciones o escritorio.

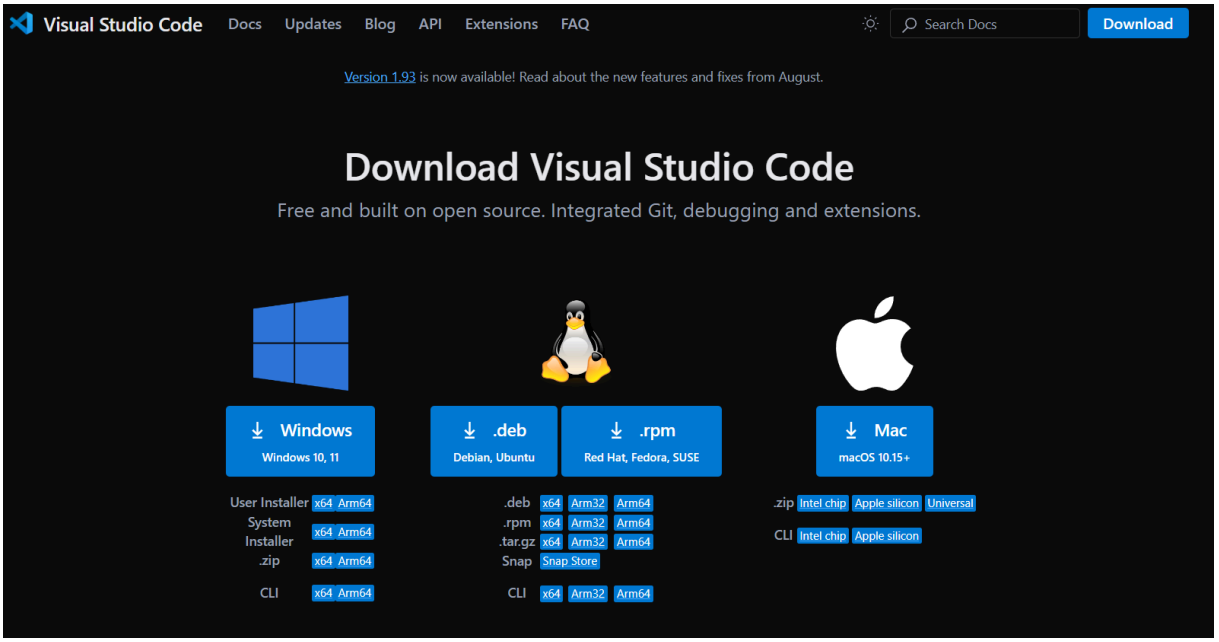

Ilustración 6.2 Instalación de Visual Studio Code

6.2.2. Manual de usuario

6.2.2.1. Descripción de la carpeta principal

Primero, detallaremos el contenido de la carpeta principal entregada para comprender el contexto y poder ejecutar la aplicación correctamente, la cual incluye los siguientes elementos:

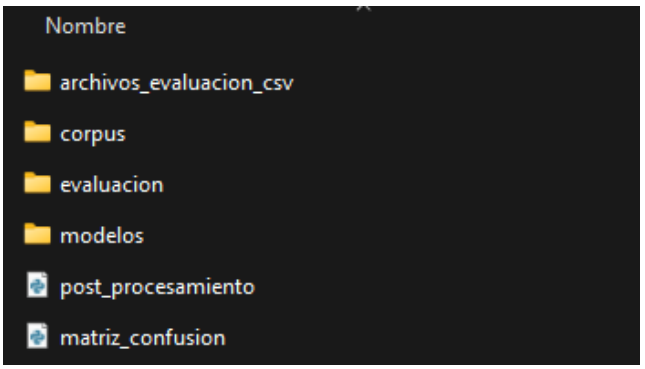

Ilustración 6.3 Carpeta principal vista desde el entorno local

- **archivos_evaluacion_csv:** esta carpeta contiene los conjuntos de datos en formato CSV que los utilizaremos para la evaluación del archivo de salida.

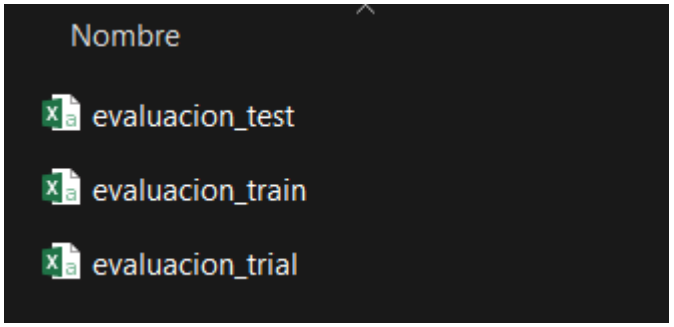

Ilustración 6.4 Contenido de la carpeta "archivos_evaluacion_csv"

- **conjunto_datos:** esta carpeta incluye los tres conjuntos de datos proporcionados. El archivo "5w1h_subtarea_2_test", relacionado con los datos de prueba, y se utilizará para predecir las etiquetas correspondientes a estos datos; "5w1h_subtarea_2_train", vinculado con los datos de entrenamiento para entrenar el modelo; y el archivo "5w1h_subtask_2_trial", que podría contiene los datos empleados para las pruebas.

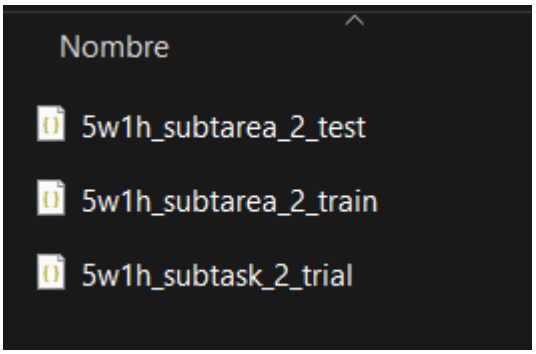

Ilustración 6.5 Contenido de la carpeta "conjunto_datos"

- **evaluacion:** esta carpeta contiene los dos archivos que nos servirán para evaluar el sistema de manera adecuada. El primero de ellos "evaluate_subtask_2", se utilizará para medir las métricas de evaluación con el archivo de prueba que nos servirá de referencia para mejorar el rendimiento del modelo. El segundo archivo "evaluation_metric_template_kaggle_subtask_2", es la plantilla que usaremos para calcular las métricas de evaluación en la competición de Kaggle de nuestro archivo test predecido.

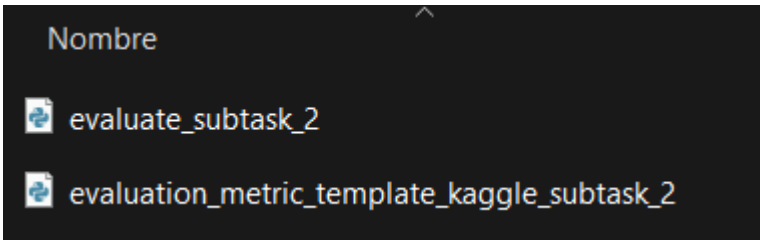

Ilustración 6.6 Contenido de la carpeta "evaluacion"

- **modelos**: esta carpeta contiene los sistemas que se han generado con las diferentes técnicas de prompt.

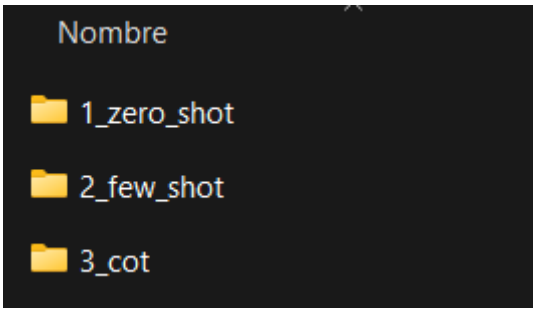

Ilustración 6.7 Contenido de la carpeta "modelos"

En cada una de estas carpetas, encontraremos la subcarpeta "train_trial", que contiene las diez pruebas distintas con diversos prompts para evaluar cuál ofrece el mejor rendimiento. Además, hay una carpeta llamada "test" que incluye los tres archivos con los mejores resultados obtenidos de la carpeta anterior, pero aplicados al archivo de test.

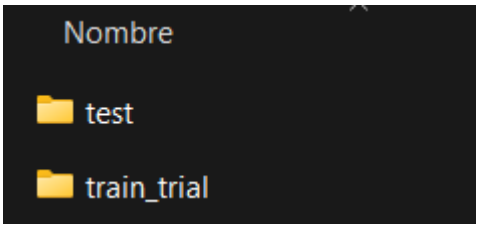

Ilustración 6.8 Contenido de la carpeta "modelos"

- **post_procesamiento:** por último este archivo corrige las etiquetas incorrectas que el modelo ha predicho y lo guarda en un nuevo archivo CSV.
- **matriz_confusion**: este fichero presenta una matriz de confusión para evaluar el rendimiento del modelo, mostrando la distribución de predicciones correctas e incorrectas.

6.2.2.2. Instrucciones de uso

Para empezar a utilizar la aplicación, es necesario abrir el entorno en el que se ejecutará, *Jupyter Notebook,* y localizar la carpeta principal entregada.

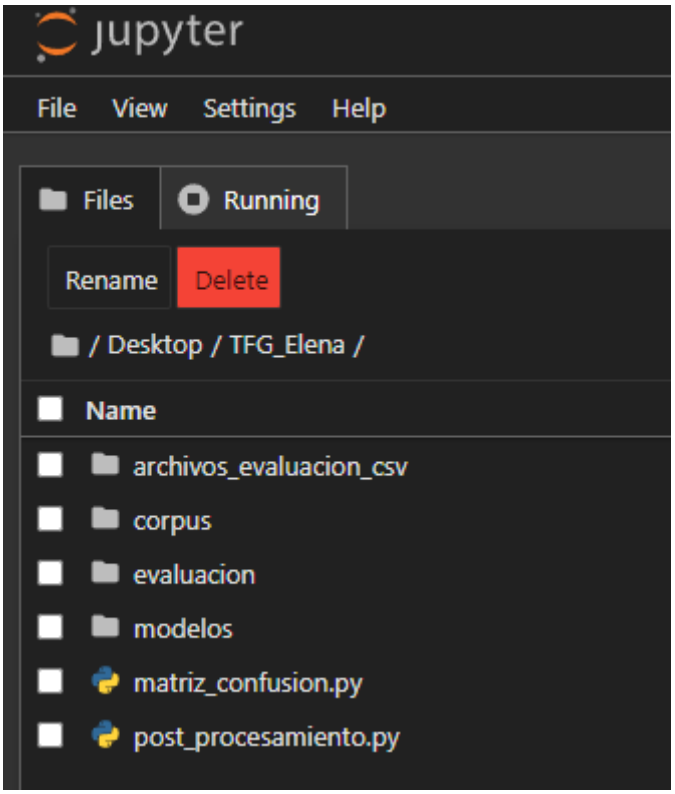

Ilustración 6.9 Carpeta principal vista desde el entorno Jupyter Notebook

Una vez ahí, accedemos a la carpeta "modelos" y seleccionamos la técnica de prompt que deseemos utilizar. Si queremos ejecutar un fichero de la carpeta "test", navegamos hasta el archivo requerido y lo abrimos en formato `**.ipynb**`.

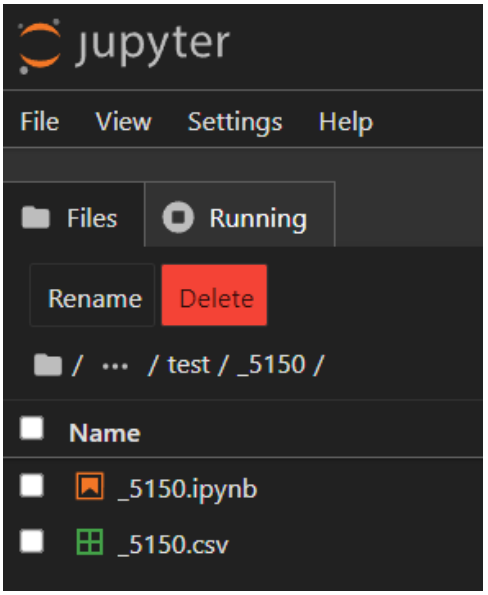

Ilustración 6.10 Contenido de la carpeta de un modelo de test

En este archivo debemos hacer un par de modificaciones en las rutas de ejecución según la técnica que hayamos seleccionado. Si estamos trabajando con la técnica “2_few_ shot" o “3_cot”, y en el caso específico de “cot_few”, necesitaremos ajustar tres rutas. Por otro lado, si estamos utilizando “1_zero_shot” o “3_cot”, en este caso “cot_zero”, solo necesitaremos modificar dos rutas.

La primera de las tres rutas que modificaremos es la del archivo de entrada, que corresponde a nuestro archivo de entrenamiento. Este fichero lo utiliza el sistema como referencia para los ejemplos, por lo que es necesario cambiar su nombre adecuadamente.

```
# Nombre del archivo de entrada
input_file = 'C:/Users/elena/Desktop/TFG_Elena/conjunto_datos/5w1h_subtarea_2_train.json'
```

Ilustración 6.11 Primera ruta a modificar (archivo de entrada)

La segunda y tercera ruta, comunes a todos los documentos, corresponden a la ubicación del archivo que se evaluará, es decir, el fichero de test y al nombre del archivo de salida donde se guardarán los resultados.

```
# Nombre del archivo de evaluación
input_file_test = 'C:/Users/elena/Desktop/TFG_Elena/conjunto_datos/5w1h_subtarea_2_test.json'
```

Ilustración 6.12 Segunda ruta a modificar (archivo de evaluación)

```
# Nombre del archivo de salida
output_file_csv= 'C:/Users/elena/Desktop/TFG_Elena/modelos/2_few_shot/test/_44/_44.csv'
```

Ilustración 6.13 Tercera ruta a modificar (archivo de salida)

No obstante, también debemos modificar las claves de los modelos generativos Gemini y ChatGPT, asegurándonos de utilizar la clave correspondiente a nuestra cuenta de usuario personal.

```
# Configuramos la API KEY de Google
GOOGLE_API_KEY='AIzaS                                    '
genai.configure(api_key=GOOGLE_API_KEY)
```

Ilustración 6.14 Clave de modelo Gemini a modificar

```
# Configuramos la API KEY de OpenAI
from openai import OpenAI
client = OpenAI(
    api_key="sk-proj-[redacted]8",
)
```

Ilustración 6.15 Clave de modelo OpenAI a modificar

Ahora, lo podremos ejecutar simplemente yendo a la barra de herramientas, al apartado “Run” y después seleccionaremos “Run All Cells’, lo que ejecutará todas las celdas del notebook.

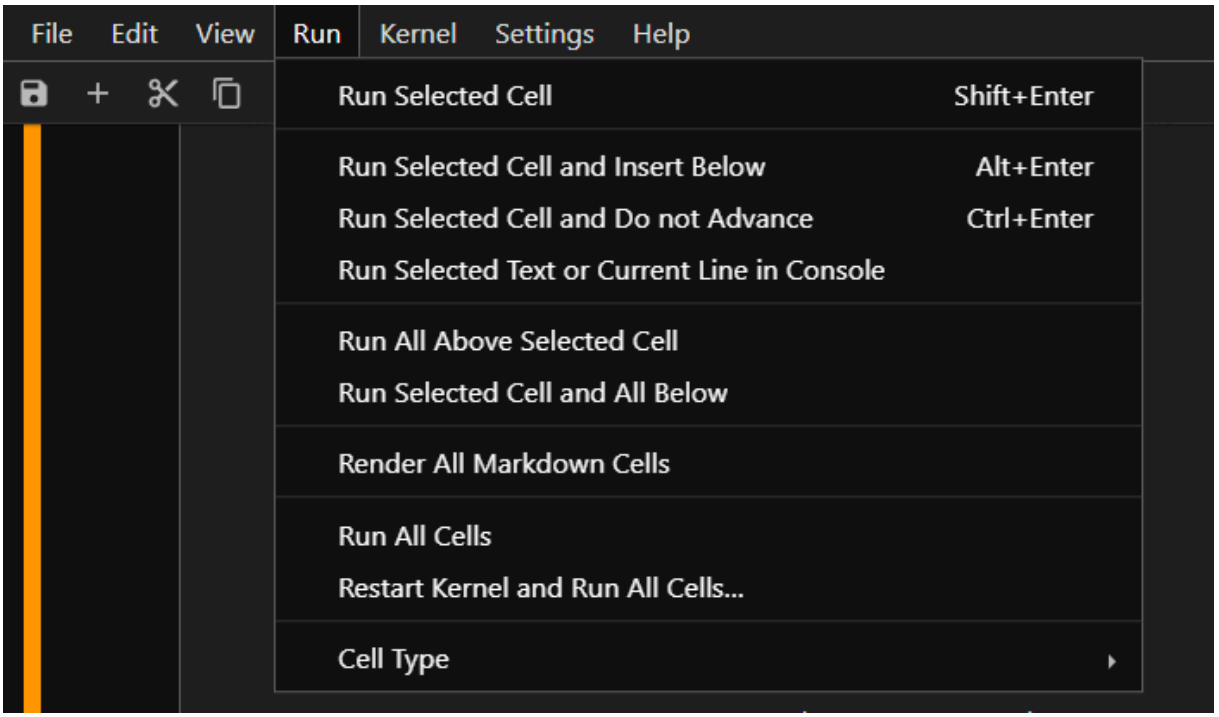

Ilustración 6.16 Cómo ejecutar todas las celdas en el entorno Jupyter Notebook

Podemos observar cómo los valores se muestran en pantalla mientras se escriben simultáneamente en el archivo CSV de salida.

Ilustración 6.17 Resultados de los valores predichos en el entorno Jupyter Notebook

En la parte superior derecha del entorno de trabajo, podemos ver el estado del kernel, las celdas ejecutadas y su tiempo de ejecución. Si el modelo está en ejecución, aparecerá como “Busy” indicando que el kernel está ocupado procesando la tarea. Una vez que ha terminado de ejecutar, el estado cambia a “Idle”, lo que significa que el kernel está libre y disponible para nuevas tareas.

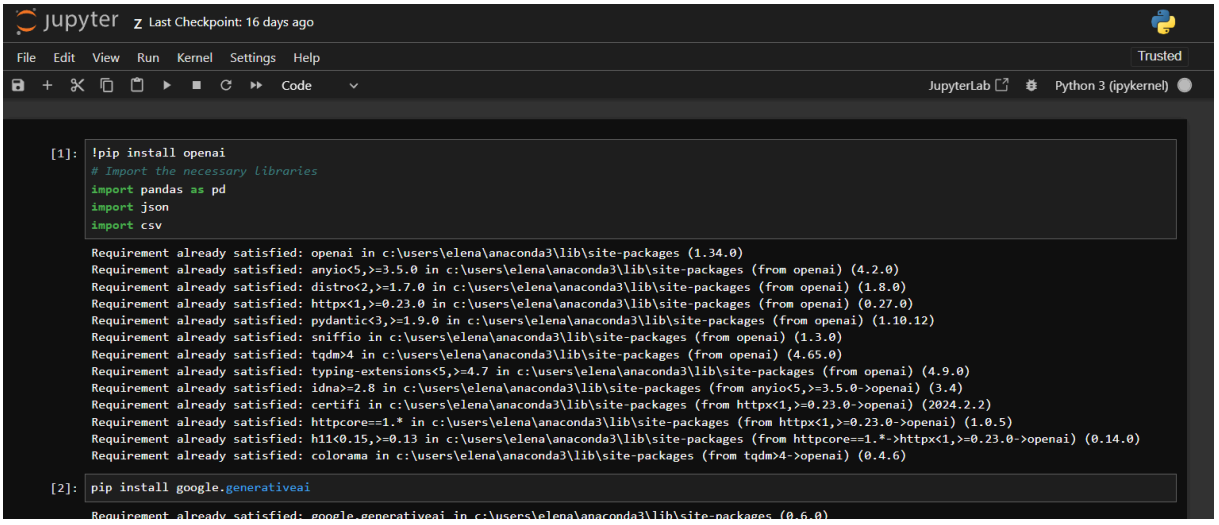

Ilustración 6.18 Entorno de Jupyter Notebook

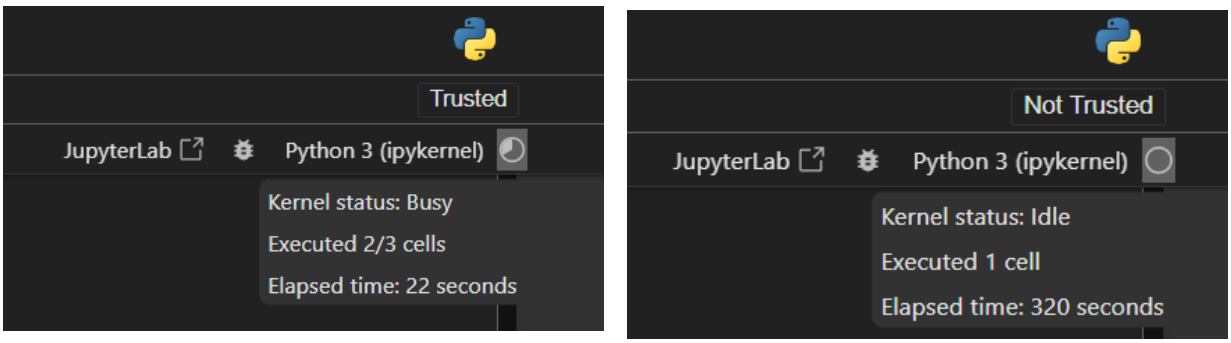

Ilustración 6.19 Estados del Kernel en el entorno Jupyter Notebook

Una vez terminada la ejecución, procederemos a evaluar el archivo de datos de salida; sin embargo, antes realizaremos un post-procesamiento. Para ello clicamos en el archivo “post-procesamiento” de la carpeta principal y lo abrimos con *Visual Studio Code*. Aquí, también deberemos cambiar las rutas correspondientes según la ubicación del archivo.

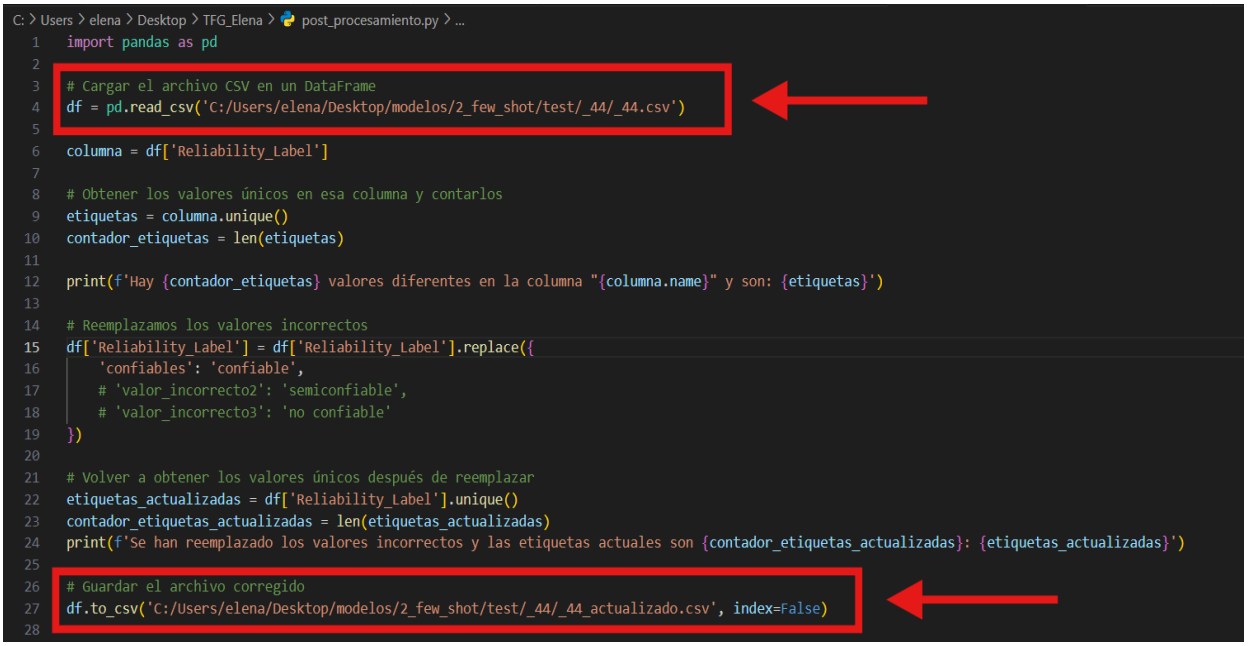

Ilustración 6.20 Rutas a modificar según la ubicación de los archivos

Al ejecutar el código, veremos que en pantalla aparecen los valores actualizados de la columna “Reliability_Label” del archivo de salida, reemplazando los valores incorrectos presentes anteriormente.

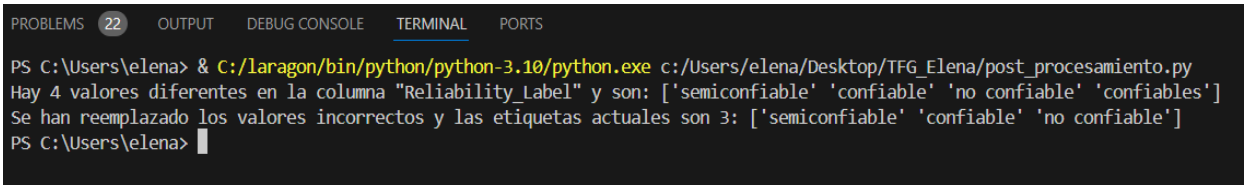

Ilustración 6.21 Resultado del archivo “post_procesamiento” por pantalla

El siguiente paso es evaluar los resultados mediante la ejecución del siguiente comando en la terminal que ejecuta el script *evaluate_subtask_2_json.py* de la carpeta “evaluacion” del proyecto proporcionando dos rutas de archivos como argumentos: la primera es el fichero con los datos correctos o “dataGold”, y la segunda es el archivo con los datos generados por el modelo.

```
Python
python evaluate_subtask_2_json.py
--pathDataGold "\archivos_evaluacion_csv\evaluacion_test.csv"
--pathDataInfered "\modelos\2_few_cot\test\_44\_44_actualizado.csv"
```

Al ejecutar el script, obtenemos las métricas de evaluación del archivo de salida generado. En este ejemplo, el resultado obtenido para el F1-score es de **0,5171**.

```
PS C:\Users\elena\Desktop> python evaluate_subtask_2_json.py --pathDataGold "\evaluacion_test.csv" --pathDataInfered "\TFG_
Elena\modelos\2_few_shot\test\_44\_44_actualizado.csv"
{'F1': 0.5171535771282441, 'Accuracy': 0.5821635012386458, 'Precision': 0.5423375482127012, 'Recall': 0.5286074605071022}
PS C:\Users\elena\Desktop>
```

Ilustración 6.22 Resultado de las métricas de evaluación del archivo ejecutado

Para identificar cómo el modelo ha categorizado cada una de las etiquetas de fiabilidad predichas, ejecutaremos el script “matriz_confusion”. Una vez abierto debemos modificar las rutas de ubicación según nuestro entorno local.

```
import pandas as pd
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Leer el archivo CSV con las etiquetas verdaderas
real_df = pd.read_csv('C:/Users/elena/Desktop/trial_sin_orden.csv')
y_true = real_df['Reliability_Label'].tolist()  # Extraer la columna 'Reliability_Label'

# Leer el archivo CSV con las etiquetas predichas
predict_df = pd.read_csv('C:/Users/elena/Desktop/modelos/2_few_shot/train_trial/vii/vii_bueno.csv')
y_pred = predict_df['Reliability_Label'].tolist()  # Extraer la columna 'Reliability_Label'

# Definir los nombres de las clases
target_names = ['confiable', 'semiconfiable', 'no confiable']

# Generar el reporte de clasificación
report = classification_report(y_true, y_pred, target_names=target_names)
print(report)

# Generar la matriz de confusión
conf_matrix = confusion_matrix(y_true, y_pred, labels=target_names)

# Visualizar la matriz de confusión
disp = ConfusionMatrixDisplay(confusion_matrix=conf_matrix, display_labels=target_names)
disp.plot(cmap='Blues', values_format='d')  # 'd' para mostrar enteros
plt.title('Matriz de Confusión (valores totales)')
plt.show()
```

Ilustración 6.23 Rutas a modificar según la ubicación de los archivos

Al ejecutarlo, nos aparecerá en pantalla una ventana emergente mostrando los valores de cada categoría que se han predicho.

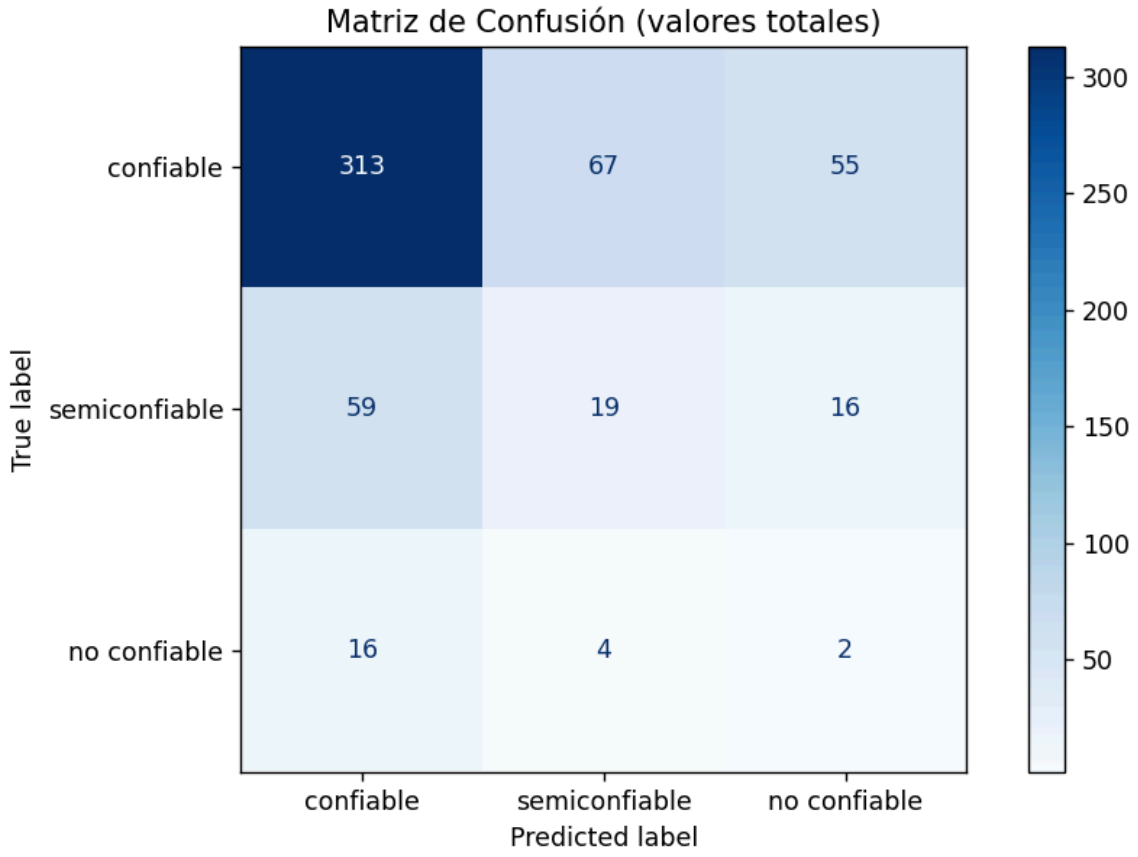

Ilustración 6.24 Resultados de la matriz de confusión

7. BIBLIOGRAFÍA

[1] *Infoxicación….* (2013, October 2). Alfons Cornella. https://alfonscornella.com/2013/10/02/infoxicacion/

[2] Romero-Rodriguez, Luis & De-Casas, Patricia & Pedreira, Mari. (2018). *Desinformación e Infoxicación en las cuartas pantallas.*

[3] *IBM. (n.d.) ¿Qué son los grandes modelos de lenguaje (LLM)?* IBM. https://www.ibm.com/es-es/topics/large-language-models

[4] Wang, M. (2024). *5W1H extraction with large language models*. ResearchGate. https://www.researchgate.net/publication/380906592_5W1H_Extraction_With_Large_Language_Models

[5] *FLARES*. (s. f.-b). https://sites.google.com/gcloud.ua.es/flares/home

[6] IberLEF Organizing Committee. (2024). IberLEF 2024: Tasks. IberLEF. https://sites.google.com/view/iberlef-2024/tasks?authuser=1

[7] Bonet-Jover, A., Sepúlveda-Torres, R., Saquete, E., Martínez-Barco, P., & Nieto-Pérez, M. (2023). *RUN-AS: a novel approach to annotate news reliability for disinformation detection*. Language Resources and Evaluation, 1-31.

[8] Vicente, M., Barros, C., Peregrino, F. S., Agullo, F., & Lloret, E. (s/f). *La generación de lenguaje natural: análisis del estado actual*. Universidad de Alicante. https://rua.ua.es/dspace/bitstream/10045/52480/1/2015_Vicente_etal_CyS.pdf

[9] Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (s/f). *Fine Tuned language models are zero-shot learners*. Arxiv.org. http://arxiv.org/abs/2109.01652

[10] Büchel, M., Li, J., & van der Herten, J. (2024). *Measuring memorization in RLHF for code completion*. ResearchGate. https://www.researchgate.net/publication/381511216_Measuring_memorization_in_RLHF_for_code_completion

[11] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, D., Batra, P., Bhargava, A., Bikel, D., Blecher, L., Boukhaled, M., Boyer, K., Bras, M., Chen, X., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Gao, H., ... & Scialom, T. (2023). *LLaMA: Open and efficient foundation language models*. arXiv. https://arxiv.org/abs/2302.13971

[12] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). *Scaling laws for neural language models*. arXiv. https://arxiv.org/abs/2001.08361

[13] Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., & Zettlemoyer, L. (s/f). *Rethinking the role of demonstrations: What makes in-context learning work?* Arxiv.org. http://arxiv.org/abs/2202.12837

[14] Prompting Guide. (n.d.). Guía de Ingeniería de Prompt. https://www.promptingguide.ai/es

[15] Rubin, V. L., Chen, Y., & Conroy, N. K. (2016). *Deception detection for news: Three types of fakes.* Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community.

[16] Pomerleau, D., & Rao, D. (2017). *Fake News Challenge* Stage 1 (FNC-I): Stance Detection. Fake News Challenge.

[17] Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). *FEVER: A Large-scale Dataset for Fact Extraction and Verification.* Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies..

[18] La atención y aumentar las visitas, L. *Cómo detectar que una noticia es falsa*. Europa.eu. https://www.europarl.europa.eu/RegData/etudes/ATAG/2017/599386/EPRS_ATA(2017)599386_ES.pdf

[19] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). *Fake news detection on social media: A data mining perspective.* ACM SIGKDD Explorations Newsletter, 19(1), 22-36. https://doi.org/10.1145/3137597.3137600

[20] IA generativa y explicable para empresas. AyGLOO. https://aygloo.com/

[21] ClaimBuster. (s/f). Uta.edu. https://idir.uta.edu/claimbuster/

[22] Zhang, A. X., Ranganathan, A., Metz, S. E., Appling, S., Sehat, C. M., Gilmore, N., ... & Mina, A. X. (2018). Una respuesta estructurada a la desinformación: Definición y anotación de indicadores de credibilidad en artículos de noticias. En *Companion Proceedings of the Web Conference* (pp. 603-612).

[23] Iberian languages evaluation forum 2024. (n.d.). Uva.Es. https://sepln2024.infor.uva.es/en/events-3/iberlef-en/

[24] *SEPLN | Sociedad Española de Procesamiento del Lenguaje Natural*. (s. f.). http://www.sepln.org/

[25] *Página de tareas aceptadas de IberLEF 2024. tasks.* (s.f.). https://sites.google.com/view/iberlef-2024/tasks?authuser=1

[26] Horne, B., & Adali, S. (2017). This just in: *Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news.* In Proceedings of the international AAAI conference on web and social media (Vol. 11, No. 1, pp. 759-766).

[27] FLARES: Subtask 2 (reliability classification). (s/f). Kaggle.com. https://www.kaggle.com/competitions/flares-subtask-2-5w1hs-classification/overview

[28] *DCY1117. (n.d.). llm-flares [Repositorio en GitHub]. GitHub. https://github.com/DCY1117/llm-flares*

[29] Piad-Morffis, A., Gutiérrez, Y., Canizares-Diaz, H., Estevez-Velarde, S., Muñoz, R., Montoyo, A., & Almeida-Cruz, Y. (2020). Overview of the ehealth knowledge discovery challenge at iberlef 2020.

[30] Zapeta Hernández, A., Galindo Rosales, G. A., Juan Santiago, H. J., & Martínez Lee, M. (2022). *Métricas de rendimiento para evaluar el aprendizaje automático en la clasificación de imágenes petroleras utilizando redes neuronales convolucionales.* https://doi.org/10.37811/cl_rcm.v6i5.3420

[31] FLARES - organizing committee. (s/f). Google.com. https://sites.google.com/gcloud.ua.es/flares/organizing

[32] *Prompt Engineering Guide: Gemini*. GitHub. https://github.com/dair-ai/Prompt-Engineering-Guide/blob/main/pages/models/gemini.en.mdx

[33] Migra a la API de Gemini desde Azure OpenAI. (s/f). Google Cloud. https://cloud.google.com/vertex-ai/generative-ai/docs/migrate/migrate-from-azure-to-gemini?hl=es-419

[34] *Extensions, L. M. A. (s/f). Visual Studio Code - code editing. Visualstudio.com.* https://code.visualstudio.com