Universidad Politécnica
de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

Análisis Comparativo de Técnicas Subsimbólicas para la Identificación de Refranes en Textos en Español

Autor(a): Luis Javier Manobanda Tutasig

Tutor(a): María del Carmen Suárez de Figueroa Baonza

Madrid, Junio, 2025

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

Trabajo Fin de Máster
Máster Universitario en Inteligencia Artificial
Título: Análisis Comparativo de Técnicas subsimbólicas para la Identificación de Refranes en Textos en Español
Julio, 2025

Autor(a): Luis Javier Manobanda Tutasig

Tutor(a):
María del Carmen Suárez de Figueroa Baonza
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

Los enunciados con sentido figurado, como los refranes son elementos de difícil comprensión por parte de ciertos colectivos (personas con discapacidad cognitiva, extranjeros con conocimientos limitados del idioma, entre otros). En la actualidad muchos de estos refranes son utilizados tanto en conversaciones informales, en medios de comunicación e incluso en entornos educativos, aportando matices y profundidad al lenguaje, y siendo estos presentes en muchos de los textos que leemos.

Los refranes al ser expresiones breves que transmiten un consejo o lección moral de forma figurada están también profundamente arraigados con el contexto cultural y frecuentemente requieren un conocimiento implícito que va más allá del significado literal del enunciado. Por otro lado, en este mismo contexto existe la ausencia de corpus que incluyan tanto refranes como sus interpretaciones, dificultando su identificación y comprensión automatizada. Siendo la mayor parte de los recursos desarrollados inicialmente para el idioma inglés, mientras que en el idioma español se cuenta con pocos estudios específicos en este ámbito.

En este proyecto se plantea realizar un análisis comparativo de técnicas subsimbólicas para la identificación de refranes en textos en español, para ello se inicia con la creación de los diferentes corpus que se utilizan para el entrenamiento, validación y prueba de los enfoques desarrollados. Entre los enfoques que se describen en este proyecto se encuentran clasificadores tradicionales (regresión logística, *random forest*, *support vector machine*), implementación de redes convolucionales para la obtención de características profundas de los refranes, así como el uso de modelos pre entrenados como FLAN-T5 y finalizando con el uso de *prompts* con el modelo GPT-4o mini. Adicional a esto se desarrolla una aplicación web que sirve para experimentar con los diferentes enfoques desarrollados en este trabajo.

La evaluación de los diferentes enfoques se realiza empleando métricas clásicas de evaluación como el *accuracy*, la precisión y *F1-Score*, además de estas métricas se hace un análisis cualitativo de los errores cometidos por los diferentes enfoques para profundizar en las causas que llevan a un enfoque a equivocarse.

Abstract

Figurative expressions, such as proverbs, are often challenging to comprehend for certain groups, including individuals with cognitive disabilities or non-native speakers with limited language proficiency. Proverbs are widely used in informal conversations, media, and even educational contexts, adding nuance and depth to language. However, these expressions, which convey advice or moral lessons in a figurative manner, are deeply rooted in cultural contexts and often require implicit knowledge beyond their literal meaning.

The lack of a comprehensive corpus containing both proverbs and their interpretations hinders the automated identification and understanding of these expressions, particularly in Spanish, where few studies address this issue. Most existing resources have been developed for the English language, leaving a significant gap in the Spanish-speaking world.

This thesis presents a comparative analysis of subsymbolic techniques for identifying proverbs in Spanish texts. It begins with the creation of various corpora for training, validation, and testing the proposed approaches. The methods explored include traditional classifiers (logistic regression, random forest, support vector machine), convolutional neural networks for extracting deep features, and the use of pre-trained models such as FLAN-T5. Additionally, a web application is developed to experiment with the different approaches implemented in this work.

The evaluation of these approaches is performed using standard metrics such as accuracy, precision, and F1-Score. Furthermore, a qualitative analysis of the errors made by each approach is conducted to explore the underlying causes of misclassification, providing deeper insights into the effectiveness of the proposed methods.

Tabla de contenido

Índice de Figuras

Índice de Tablas

1 Introducción

El lenguaje figurado, como elemento fundamental en la comunicación, representa uno de los mayores desafíos en el procesamiento del lenguaje natural (PLN o NLP por sus siglas en inglés). Se basa a menudo en el sentido común o en el conocimiento cultural compartido, y en algunos casos puede ser difícil de resolver utilizando técnicas basadas en la inteligencia artificial tanto simbólica como subsimbólica.

Su comprensión requiere no solo del entendimiento lingüístico, sino también del conocimiento cultural y del contexto social compartido, aspectos que resultan particularmente complejos de modelar mediante estadísticas lingüísticas convencionales. Esta complejidad supone un reto significativo incluso para los modelos de lenguaje (*language models* (LM) o *large language models* (LLM)) más avanzados, ya que estos, a pesar de su entrenamiento extensivo en texto, pueden carecer de la capacidad para interpretar el mundo físico, el conocimiento social o cultural en el que se basa el lenguaje [1].

La comprensión del lenguaje figurado se torna desafiante para ciertos grupos poblacionales, como las personas con discapacidad cognitiva o los hablantes no nativos del idioma [2], [3]. Esta situación plantea una importante barrera de accesibilidad que debe ser abordada para garantizar una comunicación inclusiva y efectiva. En respuesta a esta necesidad, se han desarrollado diversos métodos para la detección automática del lenguaje figurado, que van desde la extracción de características textuales [4], [5] y redes neuronales para la clasificación de textos en sentido figurado [6], [7].

Entre las distintas manifestaciones del lenguaje figurado, los refranes presentan un desafío particular debido a su naturaleza única. Siendo estas expresiones populares y tradicionales que transmiten sabiduría o consejos de manera concisa [8], los refranes están profundamente arraigados en el contexto cultural y frecuentemente requieren un conocimiento implícito que va más allá del significado literal de las palabras. En este mismo contexto existe la ausencia de corpus que incluyan tanto los refranes como sus interpretaciones dificultando aún más su detección y comprensión automatizada.

En este contexto, el presente trabajo se centra en abordar la problemática de la identificación automática de refranes en textos en español, utilizando diferentes metodologías, implementando y comparando los enfoques para determinar las estrategias más efectivas para esta tarea.

1.1 Objetivo Principal

El objetivo principal de este Trabajo de Fin de Máster es realizar un análisis comparativo de diferentes técnicas subsimbólicas para la identificación de refranes en textos escritos en español, considerando el lenguaje figurado como característica principal en los mismos.

Para alcanzar este objetivo, el trabajo se centra en abordar diferentes enfoques y evaluarlos. Dichos enfoques se basan en la extracción manual de características de los refranes, y en la recopilación de refranes para tener un corpus que se utiliza para el entrenamiento de cada modelo.

El trabajo también contempla un análisis del rendimiento de cada enfoque, considerando métricas estándar como precisión, cobertura o *recall*, y *F1-score*. Este análisis permitirá establecer recomendaciones para futuros trabajos que aborden la tarea de identificación de refranes en textos escritos en español.

Finalmente, se desarrolla una aplicación web que permite interactuar con los diferentes enfoques implementados, facilitando la comparación directa de sus resultados y proporcionando una forma práctica para la evaluación de los métodos desarrollados en un entorno real. Esta implementación no solo sirve como herramienta de validación, sino también como medio para hacer más accesible y comprensible el proceso de identificación de refranes.

1.2 Estructura del documento

El presente trabajo se estructura en cinco capítulos que abordan de manera sistemática y progresiva el desarrollo de un sistema automático para la identificación y explicación de refranes escritos en textos en español:

- El segundo capítulo, **Estado de la cuestión**, presenta una revisión de la literatura existente en el campo del lenguaje figurado, con especial énfasis en la detección automática de refranes. Se analizan diversos trabajos previos relacionados con la clasificación de textos figurativos, examinando tanto los métodos tradicionales de extracción de características textuales como los enfoques más recientes basados en modelos preentrenados. Esta revisión proporciona el fundamento teórico necesario para el desarrollo del sistema propuesto.
- En el tercer capítulo, **Metodología**, se describe los métodos de clasificación seleccionadas para el estudio comparativo. Se explica los fundamentos de cada método, incluyendo modelos tradicionales de *machine learning* y arquitecturas basadas en transformes, así como los procedimientos de limpieza y extracción de características de los datos para entrenamiento, validación y prueba.
- El cuarto capítulo, **Desarrollo**, expone en detalle la implementación técnica del sistema por cada enfoque, describiendo la arquitectura y el funcionamiento de cada componente. Se documentan las herramientas y tecnologías utilizadas, así como los procedimientos empleados en el desarrollo del sistema automático de clasificación.
- El quinto capítulo, **Resultados y Discusión**, se presenta y analiza los resultados obtenidos por cada enfoque propuesto en la clasificación de refranes en español. Se incluye un análisis comparativo de los diferentes enfoques implementados.
- Finalmente, el sexto capítulo, **Conclusiones y Líneas Futuras**, describe los hallazgos principales y las contribuciones más significativas del trabajo, además de proponer líneas de investigación futuras que podrían enriquecer y expandir los alcances del presente estudio.

2 Estado de la Cuestión

En este capítulo se realiza una revisión del estado de la cuestión de los conceptos y trabajos relacionados con la clasificación de textos en lenguaje figurado principalmente en la detección automática de refranes.

2.1 Lenguaje Figurado y Refranes

2.1.1 Lenguaje Figurado

El lenguaje figurado juega un rol en la comunicación y la cognición, siendo este último el conocimiento adquirido por la experiencia a partir de la percepción del mundo que nos rodea [1]. En la interacción cotidiana, se estima que aproximadamente el 8% de las conversaciones entre adultos incluyen expresiones irónicas, mientras que, en el ámbito educativo, los profesores recurren con frecuencia al lenguaje figurado para facilitar la enseñanza y la comprensión de conceptos abstractos.[2].

2.1.1.1 Definición

En el lenguaje figurado los significados de las palabras, oraciones y expresiones que se emplean no coinciden con su significado literal [2]. Por lo tanto, para entender el lenguaje figurado, una persona debe ser capaz de captar la intención del hablante en un contexto determinado.

2.1.1.2 Características

El lenguaje figurado abarca diversas formas de expresiones que permiten transmitir significados más allá de lo literal, enriqueciendo la comunicación y la interpretación del mensaje. Entre sus principales expresiones se encuentran la metáfora, el sarcasmo, la ironía, el símil y la sátira, cada una con características particulares que facilitan la transmisión de ideas complejas o la generación de efectos expresivos específicos.

La metáfora es un recurso lingüístico que describe un concepto en términos de otro, estableciendo una relación de semejanza implícita entre ambos [9]. No se trata de una equivalencia directa, sino de una analogía que sugiere una conexión entre los elementos comparados. Por ejemplo, en la expresión “el tiempo es oro”, se destaca el valor del tiempo a través de su asociación con un metal precioso.

El sarcasmo, por otro lado, permite al hablante expresar lo contrario de lo que realmente quiere decir, generalmente con la intención de burlarse o criticar. Se caracteriza por un tono de voz particular y el uso de expresiones que contradicen el significado literal de las palabras [7] [2]. Un ejemplo de ello sería la frase “¡Qué buen día para hacer fila en el banco!” en un contexto donde la espera es larga y tediosa.

La ironía comparte con el sarcasmo el hecho de expresar lo contrario de lo que se intenta decir, pero no siempre tiene la intención de burlarse o criticar. En este caso, la contradicción entre lo que se dice y lo que realmente se desea comunicar genera un efecto humorístico o reflexivo [7] [2]. Un ejemplo de ironía es decir “Qué buen trabajo hiciste” cuando alguien ha cometido un error evidente o exclamar “Qué suerte la mía” ante un evento desafortunado.

El símil, como la metáfora, establece una relación explicita entre dos elementos que comparten alguna característica en común, pero tiene diferencia con la metáfora, al usar nexos comparativos (“como”, “parece”, “se asemeja”, “igual que”, entre otras) para indicar la relación de semejanza [7]. Por ejemplo, “Sus ojos brillaban como estrellas en el cielo nocturno”.

Finalmente, la sátira es una forma de expresión que utiliza el humor, la ironía y el sarcasmo para criticar o ridiculizar vicios, defectos o situaciones sociales. Por ejemplo, "El gobierno anunció un nuevo plan para combatir la pobreza: reducir el número de personas consideradas pobres". En este ejemplo se utiliza la ironía para criticar la ineficacia del gobierno [7].

2.1.2 Refranes

Los refranes son una manifestación del lenguaje figurado. Se caracterizan por su conexión con la tradición y la cultura, y poseen un contenido didáctico destinado a transmitir enseñanzas, expresar verdades universales o reflejar experiencias comunes de la vida cotidiana. Para ello emplean diversos recursos del lenguaje figurado, como metáforas, símiles y otras figuras retóricas [8].

2.1.2.1 Características

Los refranes poseen diversas características que los distinguen de otras expresiones del lenguaje figurado:

- **Tradicionalidad:** los refranes son parte de la cultura que se transmite de generación en generación y su uso se asocia a un grupo social particular [8].
- **Contenido didáctico:** contienen enseñanzas morales, consejos o advertencias.
- **Forma fija:** es una característica que permite distinguir un refrán de otras expresiones del lenguaje. Debido a que los refranes tienen una estructura relativamente estable, aunque con ciertas variaciones. Siendo esta característica importante para su reconocimiento e interpretación en un discurso. Esto se debe a que tienen un significado social y psicológico mayor que otras frases, siendo esto una contribución para recordar y facilidad de reconocimiento.

2.1.2.2 Clasificación

Los refranes emplean diversas figuras retóricas para transmitir su mensaje de manera efectiva. A través de estos recursos lingüísticos, logran expresar ideas complejas de forma concisa, generando impacto y facilitando su memorización. Algunas de las principales figuras utilizadas en los refranes son:

- **Metáfora:** Consiste en trasladar el significado de una palabra a otra, estableciendo una relación de semejanza implícita. Muchos refranes utilizan metáforas para transmitir su mensaje de manera más impactante y memorable.
    - Ejemplo: "Más vale pájaro en mano que ciento volando." Aquí, "pájaro en mano" representa algo seguro y "ciento volando" algo incierto.
- **Símil**: Es una figura retórica que establece una comparación explícita entre dos elementos diferentes utilizando palabras "como" o "cual".
    - Ejemplo: "Tan cierto como que dos y dos son cuatro." Compara una verdad obvia con una certeza matemática.
- **Antítesis**: Consiste en el contraste entre dos ideas opuestas dentro de una misma expresión.
    - Ejemplo: "No hay mal que por bien no venga." Contrapone el mal con el bien que puede surgir de él.
- **Paradoja**: Se trata de una afirmación que parece contradictoria, pero que encierra una verdad subyacente.
    - Ejemplo: "El que mucho abarca, poco aprieta." Sugiere que intentar hacer demasiado puede resultar en no hacer nada bien.

2.1.2.3 Función en el lenguaje cultural

Los refranes tienen su origen en las primeras etapas de la civilización humana, lo que explica que muchos de ellos estén vinculados a contextos socioculturales e ideológicos propios de épocas pasadas. Sin embargo, la sabiduría popular y las experiencias acumuladas que transmiten mantienen su relevancia y validez en la actualidad [10]. Los refranes sintetizan en pocas palabras cientos de años de experiencia, lo que los convierte en una parte esencial del patrimonio cultural de las sociedades. Este legado refleja los elementos constantes del comportamiento humano, como la manera de enfrentar cuestiones fundamentales de la vida: el ciclo de la vida y la muerte, los sentimientos de amor y odio, las nociones de bien y mal, la búsqueda de felicidad frente al sufrimiento y las leyes que rigen tanto el mundo como las dinámicas humanas en el día a día y el trabajo [11].

2.1.3 Dificultades en la Comprensión del Lenguaje Figurado

Los refranes, como expresión del lenguaje figurado, representan una forma rica y compacta de transmitir sabiduría popular, valores culturales y experiencias acumuladas a lo largo del tiempo. Su carácter metafórico, implícito y contextual los convierte en un recurso poderoso en la comunicación, pero también en un desafío para quienes tienen dificultades para interpretar significados no literales [12].

La comprensión de los refranes exige habilidades cognitivas y sociales avanzadas, ya que su interpretación depende tanto del conocimiento cultural como del contexto en el que se utilizan [13]. Para individuos con Trastorno del Espectro Autista (TEA), trastorno específico del lenguaje o lesiones en el

hemisferio derecho del cerebro, esta tarea puede resultar especialmente complicada debido a su tendencia a interpretar el lenguaje de manera literal. Por ejemplo, un refrán como "A buen entendedor, pocas palabras" puede confundir a quienes no captan su significado implícito, relacionado con la capacidad de comprender mensajes indirectos [12].

Además, los refranes varían según la cultura y el idioma, lo que añade otra capa de complejidad. Expresiones que son claras y comunes en un contexto cultural pueden resultar opacas o incluso incomprensibles en otro. Esto refuerza la naturaleza altamente contextual y dependiente del conocimiento previo que caracteriza al lenguaje figurado en general [14]. La dificultad para interpretar los refranes no solo afecta la comprensión del mensaje, sino también la capacidad de participar plenamente en interacciones sociales, donde este tipo de expresiones suele emplearse para transmitir enseñanzas, humor o empatía. Reconocer estas barreras y fomentar estrategias de comunicación más inclusivas puede ayudar a reducir malentendidos y a promover un mayor entendimiento en contextos diversos, respetando tanto la riqueza del lenguaje figurado como las capacidades individuales [15].

2.2 Inteligencia Artificial y Procesamiento del Lenguaje Natural (PLN)

2.2.1 Introducción al Procesamiento del Lenguaje Natural

El lenguaje se define como un medio fundamental a través del cual los humanos se comunican y expresan su razonamiento. Esta capacidad se basa en la asociación de signos con significados específicos. Para establecer comunicación, el lenguaje utiliza herramientas como la escritura, las señales y la voz. En este contexto, se distinguen dos tipos principales de lenguajes [16]:

1. **Lenguaje natural**, que incluye los idiomas como el español, inglés o alemán, entre otros. Estos lenguajes evolucionan constantemente, sin adherirse estrictamente a reglas predefinidas.
2. **Lenguajes formales**, utilizados en campos como la matemática, lógica o programación, y que se rigen por reglas estrictamente definidas.

El lenguaje natural se caracteriza por su riqueza en vocabulario y construcciones, además de cualidades como la flexibilidad, la ambigüedad y la indeterminación. Estas características facilitan la comunicación humana al permitir interpretaciones variadas según el contexto. Sin embargo, representan desafíos en el procesamiento computacional, ya que dificultan tareas como el razonamiento, la caracterización y la formalización del lenguaje [17].

El Procesamiento del Lenguaje Natural (PLN) es un campo de estudio dedicado a comprender el funcionamiento del lenguaje, su estructura, la generación de nuevo lenguaje y las tareas asociadas a su tratamiento [18]. Entre las principales aplicaciones del PLN se encuentran la generación de texto, la traducción automática entre idiomas, los sistemas de preguntas y respuestas, la creación de resúmenes y el desarrollo de chatbots, entre otras. En la última

década, los avances en aprendizaje profundo han impulsado significativamente el progreso en PLN, permitiendo el desarrollo de herramientas y modelos destacados que han transformado este campo [19].

2.2.2 Algoritmos de Detección de Expresiones Figuradas

El reconocimiento de expresiones figuradas, como los refranes, representa un desafío significativo en el procesamiento del lenguaje natural debido a su naturaleza idiomática y su dependencia del contexto cultural y lingüístico. Los refranes suelen carecer de un significado literal, lo que exige que los algoritmos sean capaces de inferir interpretaciones contextuales basadas en conocimiento previo y patrones semánticos no triviales [20]. En la literatura, los enfoques para la detección de expresiones figuradas se dividen principalmente en tres categorías:

- **Reglas Basadas en Lenguaje:** Este enfoque utiliza conjuntos de reglas predefinidas que identifican expresiones figuradas mediante la correspondencia con diccionarios de refranes o patrones sintácticos recurrentes. Si bien es eficiente para idiomas con recursos bien documentados, como el español, este método es limitado por su rigidez ante expresiones variantes o reformuladas [21].

- **Modelos Estadísticos:** Los modelos estadísticos, como el *n-gram*, analizan la probabilidad de aparición de palabras en ciertas combinaciones y comparan el texto con corpora previamente etiquetados. Sin embargo, este método a menudo carece de la capacidad para capturar el significado profundo o inferir contexto cultural [22].

- **Métodos Basados en Aprendizaje Automático:** Los algoritmos de aprendizaje supervisado y no supervisado han demostrado ser más robustos en esta tarea. Modelos como *Support Vector Machines* (SVM) y *Random Forest* han sido empleados con éxito después de haber sido entrenados con características léxicas, sintácticas y semánticas. Más recientemente, el aprendizaje profundo ha permitido la detección de expresiones figuradas a través de redes neuronales recurrentes (RNN) y transformadores como BERT, que modelan relaciones contextuales profundas dentro del texto [23].

Estos algoritmos no solo permiten la detección, sino también la clasificación de las expresiones, lo que facilita su posterior interpretación y explicación en sistemas automatizados.

2.2.3 Modelos de Clasificación y Adaptación Lingüística

Una vez detectados los refranes, los sistemas de PLN deben ser capaces de clasificarlos en categorías significativas y adaptarlos para usuarios con diversas necesidades lingüísticas o cognitivas. Los modelos de clasificación y adaptación lingüística en este contexto se centran en dos objetivos principales:

- **Clasificación Temática y Funcional:** Los refranes pueden clasificarse según su contenido temático (sabiduría, advertencia, humor) o según su función en el discurso (enfatizar una idea, persuadir). Modelos como BERT y GPT-3 han sido entrenados para identificar estas características mediante métodos de *fine-tuning* en corpora específicos [23], [24]. Estos modelos aprovechan *embeddings* contextuales para asignar etiquetas significativas a cada refrán.

- **Adaptación Lingüística Inclusiva:** La adaptación de refranes busca convertir el lenguaje figurado en expresiones literales o más accesibles. Esto es particularmente relevante para personas con dificultades para comprender el lenguaje abstracto, como personas dentro del espectro autista [12]. En este ámbito, enfoques basados en redes transformadoras han mostrado eficacia. Por ejemplo, modelos generativos como T5 (*Text-to-Text Transfer Transformer*) pueden reescribir refranes en un lenguaje más directo preservando el mensaje subyacente [18].

Además, los sistemas de adaptación deben considerar la riqueza lingüística y la variación regional del español. Para abordar esto, se han integrado bases de datos multiculturales y métodos de traducción automática adaptativa que ajustan los refranes al dialecto del usuario final [25]. Estos modelos no solo mejoran la comprensión lingüística, sino que también refuerzan la accesibilidad y la inclusión en el diseño de sistemas inteligentes. Su implementación abre nuevas oportunidades en aplicaciones educativas, terapéuticas y comunicativas, marcando un avance significativo en la interacción humano-computadora.

2.3 Revisión de Modelos Existentes para la Detección de Refranes

La detección de refranes en textos es una tarea desafiante dentro del área de procesamiento del lenguaje natural (PLN), debido a la naturaleza figurada y culturalmente dependiente de estas expresiones. Aunque la literatura sobre la identificación automática de refranes es escasa, existen investigaciones y trabajos relacionados a la identificación y el procesamiento de expresiones figuradas (metáforas, ironía, sarcasmo y sátira), donde estos métodos pueden ser adaptados para la tarea de identificación de refranes. En esta sección se hablará sobre los diferentes métodos iniciando desde algoritmos basados en reglas hasta métodos avanzados de aprendizaje automático, destacando sus fortalezas, limitaciones y áreas de aplicación.

2.3.1 Algoritmos Basados en Reglas

Los enfoques basados en reglas representan una de las aproximaciones más tradicionales para la detección de refranes. Estos sistemas dependen de conjuntos de reglas lingüísticas predefinidas que identifican patrones sintácticos y léxicos característicos de los refranes. Las reglas suelen derivarse de diccionarios de expresiones figuradas o de análisis lingüístico detallado [26], [27]. En el trabajo de Rassi y colegas [28] muestra un sistema de detección de

refranes en textos escritos en portugués que utiliza reglas para la identificación de elementos centrales utilizando el etiquetado de partes del discurso (*Part of speech*) y que trabaja conjuntamente con autómatas de estados finitos logrando una precisión del 60.15%.

Estos algoritmos han dado buenos resultados en la tarea de identificación de textos, pero también presentan limitaciones y fortalezas como se describe en la Tabla 2.1.

Fortalezas	Limitaciones
• Son altamente interpretables y transparentes, lo que permite comprender cómo y por qué se detecta un refrán.	• Carecen de flexibilidad frente a variaciones lingüísticas, como parafraseos o refranes modificados.
• Funcionan bien en dominios limitados o altamente estructurados donde las expresiones son consistentes y predecibles.	• Su escalabilidad es limitada, ya que agregar nuevas reglas para diferentes dialectos o contextos puede ser laborioso.

Tabla 2.1 Fortalezas y limitaciones de los algoritmos basados en reglas.

A pesar de estas limitaciones, los algoritmos basados en reglas siguen siendo útiles en contextos donde los recursos computacionales son limitados o donde se requiere un enfoque altamente controlado.

2.3.2 Clasificadores Tradicionales de Aprendizaje Automático

Enfoques basados en clasificadores tradicionales para la identificación de textos en lenguaje figurado como SVM, regresión logística, *random forest* y árboles de decisiones (*decision trees*) han mostrado tener una mejor precisión cuando se utilizan en conjunto con características semántica y lingüísticas que han sido cuidadosamente seleccionadas [5].

2.3.2.1 Support Vector Machine

SVM es un método de clasificación binaria que busca el mejor hiperplano para dividir un conjunto de datos en dos grupos, negativo y positivo [29]. Este clasificador ha sido utilizado para la detección de lenguaje figurado en textos en ingles [30]. SVM tiene ciertas limitaciones y fortalezas como se detallan en la Tabla 2.2.

Fortalezas	Limitaciones
• Es un modelo lineal e interpretable. • Capacidad para manejar datos de alta dimensionalidad. • Son buenos manejando datos dispersos (muchas palabras raras) y con conjunto de datos escasos.	• No captura la secuencia ni el contexto lingüístico. • Entrenar con un conjunto de datos grande resulta ser muy costoso.

Tabla 2.2 Fortalezas y limitaciones de SVM.

2.3.2.2 Regresión Logística

La regresión logística es un modelo lineal que calcula la probabilidad de pertenecer una muestra a una clase específica mediante una función logística. Este modelo opera como un clasificador binario utilizando un valor umbral o de corte. Es decir, si la probabilidad calculada supera este umbral, la muestra se asigna a una clase, mientras que, si es inferior, se asigna a la otra clase. Este clasificador ha demostrado ser efectiva en diversos estudios de clasificación de lenguaje figurado en textos en inglés [5], [30], [31]. En la Tabla 2.3 se describen las fortalezas y limitaciones a considerar.

Fortalezas	Limitaciones
• Es un modelo lineal e interpretable. • Su entrenamiento es rápido incluso con muchos ejemplos. • Son buenos manejando datos dispersos y de altas dimensiones (común cuando se tiene muchas características del texto)	• No captura interacciones complejas entre palabras, por ello es necesario el uso de características adicionales. • El clasificador tiende a sobre ajustarse si se tiene características muy específicas o con pocos ejemplos.

Tabla 2.3 Fortalezas y limitaciones de la regresión logística.

2.3.2.3 Random Forest

Random Forest es un algoritmo de aprendizaje supervisado que combina múltiples estrategias: aleatorización, análisis alternativo y técnicas de conjunto. Crea un "bosque" compuesto por múltiples árboles de decisión que trabajan en conjunto. Entre sus principales fortalezas destaca su capacidad para detectar anomalías en los datos, identificar las características más relevantes y descubrir patrones complejos en los datos [32]. Se ha utilizado en diversos campos del procesamiento de lenguaje natural, como la clasificación de textos en inglés que contienen lenguaje figurado [5] y en la identificación de ironía y sarcasmo [30]. En la Tabla 2.4 se describen las fortalezas y limitaciones a considerar.

Fortalezas	Limitaciones
• Tiene una mejor capacidad para generalizar ya que utiliza múltiples árboles y promedia sus predicciones. • Maneja bien los datos con alta dimensionalidad.	• Al tener muchos arboles dificulta la capacidad de interpretación del resultado obtenido. • Entrenar es computacionalmente costoso dependiendo del número de árboles y su profundidad.

Tabla 2.4 Fortalezas y limitaciones de *random forest*.

2.3.2.4 Arboles de Decisiones

Es un algoritmo de aprendizaje supervisado utilizado en su mayor parte para tareas de clasificación como de regresión. Tiene una estructura jerárquica de árbol, compuesta por un nodo raíz, ramas, nodos internos y nodos hoja [33]. Se

ha utilizado para la identificación de lenguaje figurado en tweets escritos en ingles [5].

Estos clasificadores tradicionales dependen de la extracción manual de características léxicas, sintácticas y semánticas del texto [34], para tener una mejor precisión en su clasificación. En la Tabla 2.5 se describen las fortalezas y limitaciones a considerar.

Fortalezas	Limitaciones
• Es interpretable, se puede seguir el camino que se ha tomado para clasificar una frase. • Captura interacciones específicas entre palabras, al tener relaciones en sus ramas.	• No es eficiente con datos de alta dimensionalidad. • Se sobre ajusta cuando tienen muchas características para comparar.

Tabla 2.5 Fortalezas y limitaciones de los árboles de decisiones.

2.3.3 Modelos de Aprendizaje Profundo

Los modelos de aprendizaje profundo, en particular las redes neuronales recurrentes (RNN) y las redes neuronales convolucionales (CNN), han mejorado la capacidad de captura de patrones más complejos, incluyendo el contexto temporal y espacial de las palabras [6]. En el trabajo de Razali, M. S. B. (2023) [7] se muestra el uso de redes convolucionales como extractor de características profunda que es utilizado conjuntamente con características extraídas de forma manual para mejorar la identificación de expresiones figurativas (ironía, metáfora y sarcasmo). En la Tabla 2.6 se describen las fortalezas y limitaciones a considerar.

Fortalezas	Limitaciones
• Tienen buena capacidad para manejar la ambigüedad semántica y la dependencia de contexto. • En un contexto multilingüe se puede aprovechar datos de otros idiomas para mejorar la detección de textos en un idioma en específico.	• Se necesita grandes cantidades de datos. • Actúan como cajas negras en cuanto a su interpretabilidad. • Se pueden sobre ajustar si se tiene un corpus muy pequeño.

Tabla 2.6 Fortalezas y limitaciones de los modelos de aprendizaje profundo.

2.3.4 Modelos Basados en Transformadores

Los transformadores, como BERT (*Bidirectional Encoder Representations from Transformers*), GPT *(Generative Pre-trained Transformer) y* FLAN-T5 (*Finetuning language models Text-to-Text Transfer Transformer*), han revolucionado el campo al proporcionar *embeddings* contextuales robustos y analizar relaciones semánticas profundas en los textos [35], [36], [37]. Estos modelos permiten detectar refranes incluso cuando se presentan de forma ambigua o con variaciones lingüísticas.

Los modelos basados en transformadores han demostrado ser altamente efectivos en la detección de refranes debido a sus características avanzadas, pero también presentan ciertos desafíos como se describe en la Tabla 2.7.

Fortalezas	Limitaciones
• Capturan el contexto global del texto, lo que permite identificar refranes en oraciones complejas con significados implícitos. • Pueden adaptarse a múltiples idiomas y dialectos mediante técnicas de transferencia de aprendizaje, lo que facilita su aplicación en diversos entornos lingüísticos.	• Requieren grandes volúmenes de datos etiquetados para entrenar modelos efectivos lo que puede ser un obstáculo en lenguajes con pocos recursos. • Son computacionalmente costosos, lo que puede dificultar su implementación en sistemas con recursos limitados, como dispositivos embebidos o aplicaciones en tiempo real.

Tabla 2.7 Fortalezas y limitaciones de los modelos basados en transformadores.

2.3.5 Comparativa Entre Algoritmos Basado en Reglas y Modelos de Aprendizaje Automático

La comparación entre los enfoques basados en reglas y los modelos de aprendizaje automático destaca cómo estos últimos superan muchas de las limitaciones tradicionales (ver Tabla 2.8). Sin embargo, la elección del enfoque más adecuado depende del contexto y de los recursos disponibles [38], [39].

Aspecto	Basados en Reglas	Basados en Aprendizaje Automático
Precisión	Alta en dominios controlados	Alta en contextos diversos
Flexibilidad	Limitada	Alta
Requerimientos de Datos	Bajos	Altos
Interpretabilidad	Alta	Baja
Costo Computacional	Bajo	Alto

Tabla 2.8 Cuadro comparativo de modelos basados en reglas y basados en aprendizaje automático.

Los modelos basados en aprendizaje automático, especialmente aquellos impulsados por transformadores, son la opción preferida para sistemas modernos y escalables. Sin embargo, en escenarios donde los recursos son limitados o donde la interpretabilidad es clave, los enfoques basados en reglas continúan siendo una solución viable.

2.4 Bases de Datos y Recursos Lingüísticos sobre Refranes

El desarrollo de sistemas efectivos para la detección y explicación de refranes requiere el acceso a bases de datos lingüísticas bien diseñadas y recursos que contengan información cultural y semántica sobre estas expresiones.

2.4.1 Bases de Datos Disponibles

Entre los recursos más utilizados en el ámbito académico y comercial se encuentran colecciones de refranes organizadas por idioma, región y significado. Entre los recursos disponibles se encuentran:

- **Proverbia** [40]**:** Una extensa base de datos de refranes clasificados por temática y contexto cultural, que permite identificar patrones frecuentes en diferentes idiomas.

- **Centro Virtual Cervantes** [41]**:** Una fuente especializada en refranes y expresiones idiomáticas del español, ampliamente utilizada en investigaciones lingüísticas.

- ***Open Multilingual WordNet*** [42]**:** Un recurso léxico multilingüe que incluye equivalentes de refranes en varios idiomas.

2.4.2 Necesidades y Retos de las Bases de Datos Disponibles

Entre los principales retos y necesidades para disponer de una base de datos accesible y con características relevantes para la tarea de identificación de refranes se consideran los siguientes:

1. **Cobertura Cultural:** Los refranes varían ampliamente entre regiones y culturas, lo que hace necesario contar con bases de datos que reflejen esta diversidad [43].

2. **Anotaciones Semánticas:** Para tareas de NLP, es fundamental disponer de anotaciones semánticas y contextuales que ayuden a los modelos a entender no solo la estructura lingüística, sino también el significado figurado [44].

3. **Actualización Dinámica:** Los refranes evolucionan con el tiempo, y su uso puede variar dependiendo de las tendencias sociales. Por tanto, es crucial que los recursos lingüísticos sean actualizables de manera dinámica [45].

2.4.3 Uso en el Entrenamiento de Modelos

Las bases de datos de refranes se utilizan para entrenar modelos de aprendizaje automático, proporcionando ejemplos etiquetados que los algoritmos pueden analizar y generalizar. Estas bases de datos también permiten evaluar la capacidad de los sistemas para identificar y explicar refranes en textos complejos. Por lo que estos recursos lingüísticos al ser correctamente diseñados se convierten en la piedra angular de cualquier iniciativa orientada a la detección y explicación automática de refranes, y su desarrollo continuo es esencial para el avance de esta área.

2.5 Métricas de Evaluación

La evaluación de modelos para la detección de refranes es un aspecto crítico en el desarrollo de sistemas eficientes y confiables. Este proceso implica medir la capacidad de los algoritmos para identificar, clasificar y, en algunos casos, explicar el significado de los refranes en diferentes contextos. A continuación, se analizan las métricas más relevantes y su aplicación en este campo.

2.5.1 Métricas Clásicas

Las métricas clásicas, ampliamente utilizadas en tareas de procesamiento del lenguaje natural (PLN), son la base para evaluar la calidad del desempeño de los sistemas de detección de refranes:

1. **Precisión (*Precision*):** Define la proporción de refranes detectados correctamente entre todos los identificados por el modelo. Es una métrica crucial en contextos donde los falsos positivos (como expresiones incorrectamente clasificadas como refranes) pueden afectar la utilidad práctica del sistema, como en aplicaciones educativas o contextos inclusivos [46].
    - **Fórmula:**

$$Precisión = \frac{Verdadero\ Positivo}{Verdadero\ Positivo\ +\ Falso\ Positivo} \quad (2.1)$$

    - **Aplicación:** Ideal en tareas donde la precisión es más importante que la exhaustividad, por ejemplo, al proporcionar explicaciones detalladas de refranes.

2. **Exhaustividad (*Recall*):** Mide la proporción de refranes correctamente identificados entre todos los refranes reales presentes en un texto. Es fundamental en aplicaciones donde la omisión de refranes puede tener consecuencias negativas, como en análisis culturales o educativos [46].
    - **Fórmula:**

$$Exhaustividad\ = \frac{Verdadero\ Positivo}{Verdadero\ Positivo + Falso\ Negativo} \quad (2.2)$$

- **Aplicación:** Resulta crítica cuando la prioridad es capturar todos los refranes posibles, incluso a costa de detectar falsos positivos.

3. **Medida F1 (*F1-Score*):** Es la media armónica de la precisión y la exhaustividad, lo que proporciona un balance entre ambas métricas.
    - **Fórmula:**

$$F_1 score = 2 \times \frac{Precisión \times Exhaustividad}{Precisión + Exhaustividad} \quad (2.3)$$

    - **Aplicación:** Es especialmente útil en escenarios con desequilibrios en las clases (es decir, textos con pocos refranes en comparación con otras expresiones) [46].

2.5.2 Métricas Avanzadas

A medida que los sistemas para la detección de refranes han evolucionado, han surgido métricas avanzadas que evalúan aspectos más complejos, como el contexto y la semántica de los refranes:

1. **Exactitud de Contexto:** Evalúa si el modelo puede identificar correctamente un refrán en el contexto adecuado, considerando la ambigüedad inherente de las expresiones figurativas. Por ejemplo, un refrán puede tener múltiples interpretaciones dependiendo del entorno lingüístico o cultural en el que aparece.
    - **Desafío:** Esta métrica requiere conjuntos de datos ricos en anotaciones contextuales.

2. **Medida de Similaridad Semántica:** Utiliza técnicas como la *cosine similarity* entre *embeddings* de texto para medir qué tan bien el modelo comprende el significado figurado del refrán en comparación con su interpretación esperada [47].
    - **Aplicación:** Es clave para evaluar modelos basados en transformadores como BERT o GPT, que dependen de representaciones semánticas profundas.
3. **Evaluación de Parafraseo:** Determina si el modelo puede reconocer variaciones o parafraseos de refranes, algo crucial en escenarios reales donde estas expresiones pueden no aparecer de forma exacta [48], [49].
    - **Herramientas:** Técnicas como BLEU (*Bilingual Evaluation Understudy*) o ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) son útiles para medir la similitud entre refranes detectados y su representación esperada.

2.5.3 Evaluación de Explicabilidad

Más allá de la detección, algunos sistemas buscan explicar el significado de los refranes, lo que requiere métricas específicas para evaluar la calidad de estas explicaciones.

1. **Claridad:**
   Mide qué tan comprensible es la explicación generada para un usuario promedio, especialmente aquellos con capacidades cognitivas o lingüísticas diversas [50].
   - **Métodos:** Encuestas a usuarios y métricas de legibilidad, como el índice Flesch-Kincaid [51].

2. **Coherencia:** Evalúa si la explicación generada es lógica y consistente con el significado del refrán.
   - **Aplicación:** Útil en sistemas educativos y asistentes virtuales que buscan enseñar el uso y la interpretación de refranes.

3. **Pertinencia:** Determina si la explicación se ajusta adecuadamente al contexto en el que aparece el refrán, asegurando que no haya una desconexión entre la interpretación ofrecida y el uso real del refrán [50].

3 Metodología

Este capítulo presenta la metodología desarrollada para la identificación de refranes en textos escritos en español. En la Figura 3.1 se muestran los procedimientos, métodos y enfoques implementados para abordar la tarea de identificación de refranes.

En primer lugar, se detalla el proceso de recopilación y creación del corpus de textos, que incluye tanto refranes como oraciones que no son refranes. Debido a la ausencia de un corpus de refranes en español accesible, estos se obtuvieron de diferentes fuentes confiables considerando la variabilidad de estos y que contengan su significado para cada refrán. Como siguiente paso se considera el preprocesamiento de los datos considerando si es necesario eliminar signos de puntuación u otro signo que son propios de los refranes. Además, se realizó un análisis de las características distintivas de los refranes para tener un mejor conocimiento de las propiedades de los refranes.

Otro aspecto importante por considerarse es la representación de los textos que se van a utilizar para la tarea de identificación de refranes, si bien estos pueden ser representados mediante *embeddings* se consideró también el uso de otras características que pueden ser extraídas de forma manual con el fin de capturar características propias de los refranes. Como paso siguiente se ha diseñado los diferentes enfoques para la identificación de refranes, iniciando desde la selección de clasificadores tradicionales como regresión logística, *random forest* y SVM en combinación con redes neuronales convolucionales hasta enfoques utilizando modelos preentrenados y *prompting*.

Finalmente, y como objetivo de este trabajo de fin de Máster se determina las métricas a utilizar para la evaluación de los diferentes enfoques utilizados en este trabajo, así como también considerar un análisis cualitativo de los errores más comunes y recurrentes de los diferentes enfoques con el fin de poder entender el comportamiento de los métodos de clasificación implementados y comprender las causas que llevan a un enfoque a equivocarse.

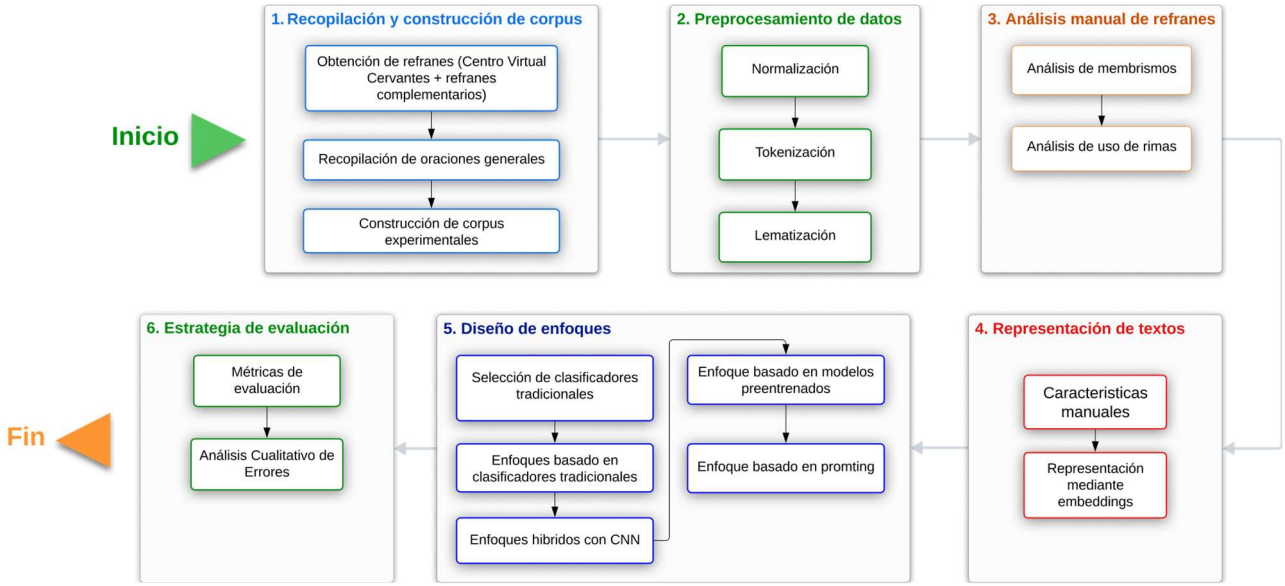

Figura 3.1 Diagrama de flujo de la metodología utilizada para la identificación de refranes.

3.1 Construcción de Corpus para la Tarea de Identificación de Refranes

Para abordar la tarea de identificación de refranes es necesario tener datos tanto de ejemplo positivos (refranes) como negativos (oraciones generales) que serán utilizados para el entrenamiento, validación y prueba de los distintos métodos de identificación de refranes.

3.1.1 Selección de Fuentes de Refranes

Dos fuentes principales se han utilizado para obtener los refranes, cada una aportando características y elementos distintos para enriquecer nuestro conjunto de datos, siendo estos del Centro Virtual de Cervantes [41] y del Diccionario de Refranes de José María Sbardi y Osuna [52].

3.1.1.1 Refranes del Centro Virtual Cervantes

El Centro Virtual de Cervantes [53] proporciona un refranero en español agrupado alfabéticamente. Esta base de datos no solo proporciona el enunciado del refrán en sí, sino que también incluye información de sus características lingüísticas y contextuales. La Tabla 3.1 se muestra un ejemplo de las características que proporciona el Centro Virtual de Cervantes para el refrán “A caballo regalado, no le mires el diente”.

Característica	Dato	Descripción
Enunciado	A caballo regalado, no le mires el diente	
Ideas claves	Educación, Apreciación, Conformidad	Conceptos fundamentales que capturan la esencia del refrán.
Significado	Este refrán recomienda aceptar los regalos de buen grado y sin poner reparo alguno, pues se considera descortés el analizar exhaustivamente la calidad del obsequio, así como resaltar sus defectos o fallos.	Explicación del mensaje o enseñanza que transmite el refrán.
Marcador de uso	Muy utilizado	Indica la vigencia y frecuencia de uso del refrán.
Observaciones léxicas	En esta paremia, diente equivale a «dentadura». La voz «caballo» es muy frecuente en las paremias, pues este animal fue uno de los principales medios de transporte hasta el siglo XIX, siglo en que empezó a disminuir su importancia con la llegada del automóvil y su paulatina extensión.	Análisis lingüístico del refrán.
Fuentes	TERREROS Y PANDO, E. de (1786-1793 = 1987): Diccionario castellano con las voces de ciencias y artes. Edición facsímil, Madrid, Arco/Libros.	Descripción de donde se hace referencia el uso del refrán.
Observaciones	El referente de este refrán se encuentra en las ferias de ganado, en las que el comprador comprueba la edad y la salud del caballo por el estado de su dentadura.	Notas adicionales sobre el refrán
Variantes	• A caballo regalado, no hay que mirarle el diente • A caballo presentado, no hay que mirarle el diente	Posibles formas que hace referencia al mismo refrán.
Sinónimos	• A quien dan, no escoge	Refranes que expresan una idea similar.
Antónimos		Refranes que expresan una idea opuesta.
Contextos	«Sin que yo, por mi parte, la haya solicitado, ni poder explicarme por dónde me ha venido, me he encontrado con la vida; y como suele decirse que a caballo regalado no hay que mirarle el diente, sin discutirla, sin analizarla, me limito a sacar de ella el mejor partido posible» (Gustavo Adolfo Bécquer, Memorias de un pavo [Narraciones]. Madrid: Turner, 1865=1995, p. 326).	El contexto en donde se hace uso el refrán.

Tabla 3.1 Ejemplo de las características de un refrán obtenido del Centro Virtual Cervantes [53]

El corpus recopilado será referido en adelante como “Corpus Cervantes” en este Trabajo Fin de Máster. Es necesario mencionar que la complejidad de información varía por cada refrán, pero en su mayor parte mantiene una

estructura básica que comprende el enunciado del refrán y su correspondiente significado.

3.1.1.2 Fuentes Complementarias

Adicionalmente, se considera como fuente complementaria el Diccionario de Refranes de José María Sbardi y Osuna [52] (referido en adelante como "Corpus Sbardi"). Este diccionario sirve para enriquecer la variedad de refranes y aumentar la dimensión cultural inherente de los mismos, aspecto que se menciona en la sección 2.1.2. Esta colección de refranes complementarios presenta una estructura más concisa limitándose al enunciado del refrán y su significado.

3.1.2 Selección de Textos que no son Refranes

Para la tarea de identificación de refranes es necesario tener textos que no son refranes, por esta razón para completar nuestro conjunto de datos con ejemplos negativos (oraciones que no son refranes) se ha recopilado datos mediante tres aproximaciones complementarias, cada una diseñada para aportar diferentes tipos de contraejemplos con el objetivo de mejorar la capacidad discriminativa de los diferentes métodos de clasificación.

3.1.2.1 Oraciones Generales

La primera aproximación utilizada consistió en la recopilación de oraciones generales de dos fuentes:

1. **Corpus Ancora [58]:** que es una recurso existente y accesible que contiene una amplia variedad de oraciones.

2. **Oraciones de cuentos:** un conjunto de oraciones, proporcionadas por el *Ontology Engineering Group* (OEG) del Departamento de Inteligencia Artificial (DIA) de la UPM. La inclusión de estos fragmentos de cuentos corresponde a la consideración de que estos textos suelen presentar esquemas rítmicos similares a los refranes [59].

La combinación de estos dos conjuntos de datos da origen al "Corpus Oraciones".

3.1.2.2 Trigramas a partir de Refranes

Como segunda estrategia y para reforzar la capacidad discriminativa de los diferentes métodos de identificación de refranes que se abordan en este trabajo, se ha decidido agregan ejemplos negativos a partir de los trigramas de cada refrán para evitar que estos aprenda de combinaciones comunes de los refranes, con el fin de reducir el sesgo hacia patrones léxicos comunes en refranes que podrían llevar a falsos positivos.

Estas secuencias de trigramas tienen como objetivo evitar causar una clasificación errónea como refrán, cuando en realidad es una expresión literal, obligando a los métodos a identificar refranes y evitar que estos memoricen coincidencias superficiales de palabras. A este conjunto de datos se lo conocerá en adelante como “Corpus Trigramas”.

3.1.2.3 Oraciones Literales a partir de Trigramas

En esta última estrategia, se pretende reforzar la discriminación semántica de los métodos de clasificación, construyendo un corpus adicional utilizando los trigramas identificados previamente.

Estas oraciones representan usos literales de las mismas combinaciones de palabras que aparecen en los refranes, proporcionando así contraejemplos para el entrenamiento del modelo. Este corpus se lo conocerá como “Corpus Oraciones de Trigramas”.

3.1.3 Resumen de los Corpus Iniciales

En la siguiente Tabla 3.2 se muestran los corpus obtenidos, indicando su origen y el propósito en el entrenamiento de los métodos de clasificación.

Nombre	Origen	Descripción
Corpus Cervantes	Centro Virtual Cervantes [41]	Refranes en español con información detallada de sus características lingüísticas y contextuales.
Corpus Sbardi	Diccionario de Refranes de José María Sbardi y Osuna [52]	Refranes complementarios con estructura simplificada (enunciado y significado).
Corpus Oraciones	Corpus Ancora [54]	Oraciones que no son refranes, incluyendo fragmentos de cuentos.
Corpus Trigramas	Generado a partir del Corpus Cervantes	Secuencias de trigramas (tres palabras extraídas de los refranes)
Corpus Oraciones de Trigramas	Generado usando API de OpenAI	Oraciones construidas usando los trigramas identificados en contextos literales.

Tabla 3.2 Corpus iniciales obtenidos de los diferentes recursos.

3.2 Construcción de Corpus Experimentales

Para este Trabajo de Fin de Máster, se han desarrollado diferentes combinaciones de los corpus iniciales previamente descritos como se muestra en la Tabla 3.3. Estas combinaciones de corpus se han realizado para entrenar los diferentes modelos de identificación de refranes y evaluar el impacto de las diferentes combinaciones de corpus.

Nombre	Corpus que intervienen	Descripción
Corpus 1	Corpus Cervantes Corpus Oraciones	Su característica distintiva es la lematización completa de todos los textos, donde cada palabra ha sido reducida a su forma base. Esta transformación lingüística permite a los modelos identificar patrones estructurales de los refranes independientemente de las variaciones morfológicas del español. Es el corpus más pequeño de todos.
Corpus 2	Corpus Cervantes Corpus Oraciones	Sus datos no han sido lematizados (siendo el único corpus con esta característica) con el objetivo de evaluar el impacto de la lematización en los modelos, ya que este proceso elimina la rima, una característica distintiva de los refranes [59].
Corpus 3	Corpus Cervantes Corpus Sbardi Corpus Trigramas Corpus Oraciones	Incorpora ejemplos negativos basados en trigramas.
Corpus 4	Corpus Cervantes Corpus Sbardi Corpus Oraciones de Trigramas Corpus Oraciones	Utiliza oraciones construidas a partir de trigramas.
Corpus 5	Corpus Cervantes Corpus Sbardi Corpus Trigramas Corpus Oraciones de Trigramas Corpus Oraciones	Combina todos los recursos disponibles, siendo el corpus más extenso de todos.

Tabla 3.3 Corpus experimentales utilizados para el entrenamiento, validación y test.

Para este trabajo, los corpus se han dividido en tres conjuntos siguiendo las prácticas estándar en aprendizaje automático: entrenamiento (80%), validación (10%) y prueba (10%).

3.3 Preprocesamiento de Datos

Para la limpieza de los datos obtenidos se ha usado una normalización y preprocesamiento de los mismos, siendo la normalización convertir el texto en minúsculas y eliminar caracteres especiales manteniendo signos de puntuación que son propios de los refranes, esta consideración se ha tomado debido a que un refrán se puede dividir en bimembre o plurimembre [55] [56] [57], donde se utiliza los signos de puntuación para dividir un refrán en versos. Mientras que en el preprocesamiento se eliminan los signos de puntuación, se lematiza el texto y se eliminan los *stop word* (o palabras vacías) que son palabras que aparecen frecuentemente en el idioma español y que aportan poco valor

semántico por sí solas. En la Figura 3.2 se muestra en detalle el proceso de normalización y de preprocesamiento, siendo la normalización la eliminación de caracteres especiales (se mantiene los signos de puntuación) que se encuentran en los textos, mientras que el preprocesamiento utiliza el texto limpio producto de la normalización para luego lematizarlo y eliminar los *stop words*, este proceso si considera la eliminación de los signos de puntuación.

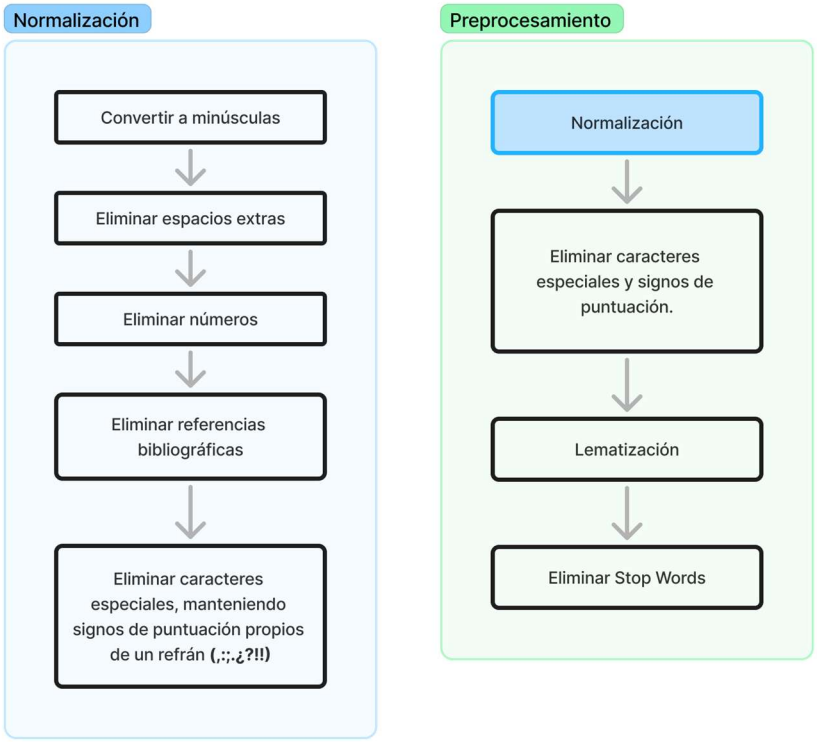

Figura 3.2 Normalización y preprocesamiento de datos

El preprocesamiento permitirá evaluar el impacto de la lematización en los métodos de clasificación, ya que este proceso elimina la rima, una característica distintiva de los refranes [55]. Esta distinción permitirá analizar cómo afecta la presencia o ausencia de la rima en el rendimiento de los diferentes enfoques que se detallan en secciones posteriores.

3.4 Análisis Manual de Refranes

El refrán es una frase completa e independiente que tiene un sentido directo o preferentemente figurada que expresa un sentimiento, hecho, enseñanza a manera de juicio. Y su estructura incorpora al menos dos ideas relacionadas entre sí para expresar una enseñanza, sentido o hecho a manera de juicio [55].

Por su naturaleza folclórica, los refranes son un reflejo del conocimiento popular que contiene costumbres y saberes ancestrales [56]. A partir del análisis de 12.127 refranes provenientes de los Corpus Cervantes y Sbardi, se ha identificado un conjunto de palabras que aparecen con alta frecuencia. La representación visual de estas frecuencias mediante una nube de palabras

permite observar los términos más utilizados en los refranes en español, como se muestra en la Figura 3.3.

Figura 3.3 Nube de palabras de los Corpus Cervantes y Sbardi.

Otra característica para considerar es la brevedad de los refranes, como se muestra en la Figura 3.4, un histograma que representa la distribución de la cantidad de palabras por refrán. El eje X indica el "Número de palabras" que va de 0 a 38 palabras y el eje Y muestra la "Frecuencia". Mostrándonos que la mayoría de los refranes contienen entre 6 y 8 palabras, con un pico notable alrededor de 7 palabras, siendo también poco frecuentes los refranes con menos de 4 palabras o más de 15 palabras. Este histograma muestra los refranes sin eliminar *stop words*.

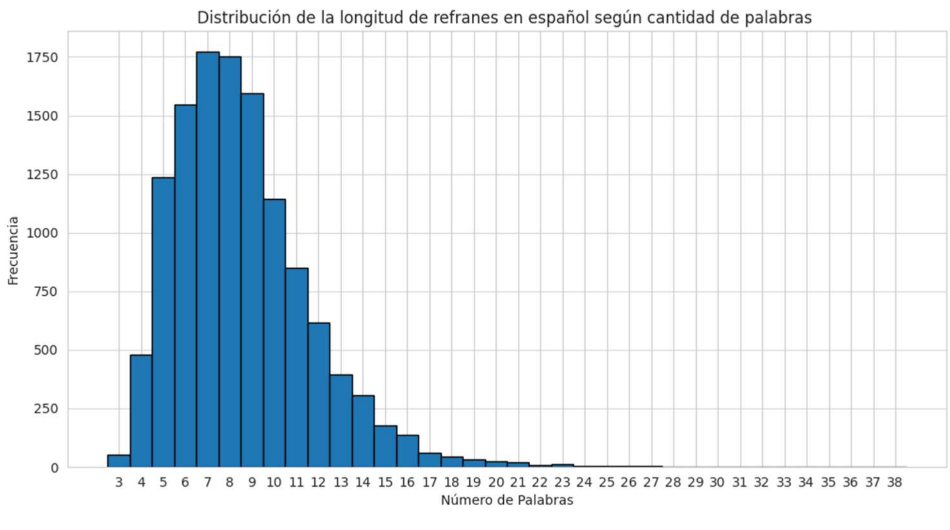

Figura 3.4 Distribución de la longitud de refranes en español según cantidad de palabras del Corpus Cervantes y Sbardi.

Por otro lado, si se remueven los stop *words* que son palabras que son muy comunes en el idioma español y no aportan un valor semántico, se observa que el rango de palabras por refrán va de 3 a 17 palabras, y la frecuencia más alta son refranes con 4 palabras, siendo muy poco frecuentes los refranes con 8 palabras, como se muestra en la Figura 3.5. Esta distribución indica que, al

eliminar los stop *words*, la mayoría de los refranes en español quedan compuestos por 3 a 6 palabras significativas.

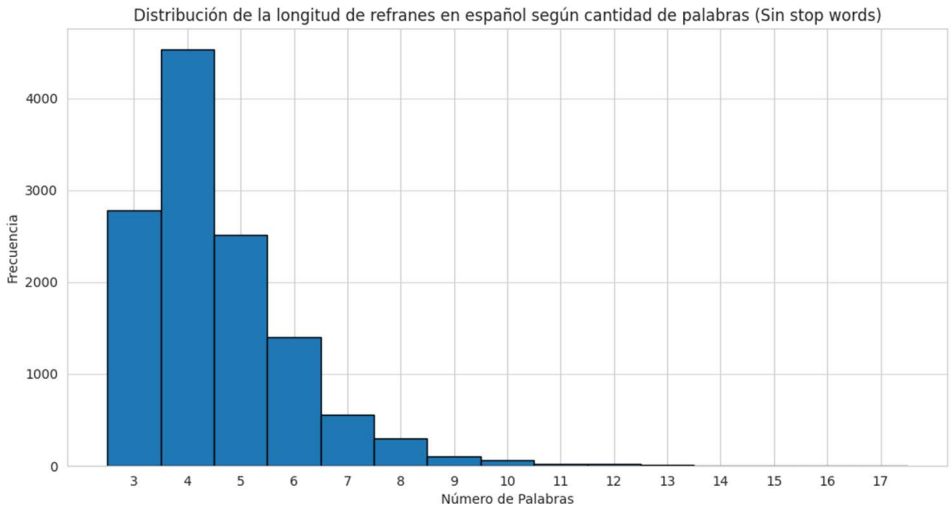

Figura 3.5 Distribución de la longitud de refranes en español según cantidad de palabras (Sin *stop words*).

La diferencia con el histograma anterior (sin remover *stop words*) es notable, ya que muestra cómo los *stop words* contribuyen a la longitud total de los refranes, pero no necesariamente a su contenido semántico.

En lo que corresponde a su sentido semántico del refrán se refiere al significado real o el sentido que transmiten las palabras en el refrán, por ejemplo:

El refrán “Amores reñidos son los más queridos”:

- Contando sus palabras se tiene un total de 7 palabras: “Amores”, “reñidos”, “son”, “los”, “más”, “queridos”.
- Removiendo *stop words* se tiene un total de 4 palabras: “amores”, “reñidos”, “queridos”.

Las palabras eliminadas (“son”, “los”, “más”) son *stop words*. Siendo las palabras que quedan (“amores”, “reñidos”, “queridos”) las que contienen el contenido semántico, y transmiten el mensaje principal del refrán.

Por esta razón cuando el histograma muestra que la mayoría de refranes tienen 4 palabras después de eliminar *stop words*, nos indica que los refranes en español tienden a transmitir sus enseñanzas o mensajes usando un número relativamente pequeño de palabras con contenido significativo, lo que contribuye a que sean fáciles de recordar y transmitir [56].

3.4.1 Análisis de Membrismos en los Refranes

Los refranes tienen presencia de rasgos unimembres, bimembres (que se componen de dos sentencias) o en algunos casos plurimembres, pero en la mayoría de los refranes se presentan dos ideas.

Para extraer estos miembros de un refrán se pueden usar los signos de puntuación que dividen a un refrán en sus miembros, como se muestra en la Tabla 3.4.

Refrán	Miembros
A buen servicio, mal galardón	"A buen servicio", "mal galardón"
A cada olla, su cobertera	"A cada olla", "su cobertera"
A cada pajarillo le gusta su nidillo	"A cada pajarillo le gusta su nidillo"
A la noche, arreboles; a la mañana habrá soles	"A la noche, arreboles", "a la mañana habrá soles"

Tabla 3.4 Ejemplo de refranes divididos en miembros por sus signos de puntuación.

Analizando el Corpus Cervantes y Sbarbi se obtiene la Tabla 3.5, que muestra que existe una mayor cantidad de refranes que son unimembres y bimembres.

Número de miembros	Número de refranes
1	5146
2	5643
3	891
4	356
5	45
6	25
7	9
8	7
9	2
10	1
11	1
12	1

Tabla 3.5 Número de miembros en los refranes de Cervantes y Sbarbi

3.4.2 Análisis del Uso de Rima en los Refranes

La rima es una de las características principales de los refranes, ya que suelen estar compuestos por dos versos, que pueden ser desiguales y presentar rima asonante, manteniendo una estrecha relación con la poesía [56]. Además, como

se mencionó en la sección 3.4.1, la mayoría de los refranes tienen una estructura bimembre con una pausa intermedia que separa dos cláusulas rimadas. Asimismo, emplean diversos recursos estilísticos como el metro, la aliteración, el paralelismo, la similicadencia y el dialogismo [55], [57].

El análisis de los Corpus Cervantes y Sbarbi revela patrones interesantes en la distribución de palabras rimadas. Como se muestra en la Tabla 3.6, la mayoría de los refranes (6.274) contienen dos palabras que riman entre sí, lo que refuerza la estructura bimembre mencionada. Un número significativo de refranes (4.753) no presenta rimas, mientras que es menos común encontrar refranes con tres o más palabras rimadas. La frecuencia disminuye notablemente a medida que aumenta el número de palabras que riman, siendo extremadamente raros los casos con más de 6 palabras rimadas.

Número de palabras que riman	Número de refranes
0	4753
2	6274
3	436
4	535
5	82
6	34
7	8
8	3
9	2

Tabla 3.6 Número de palabras que riman en los Corpus Cervantes y Sbarbi.

3.5 Representación de Textos para la Clasificación

En lo que corresponde a la tarea de identificación de refranes es necesario tener una representación de los textos. En este trabajo se ha considerado dos enfoques para representar los textos, siendo estos: la extracción manual de características y la representación mediante *embeddings*. Estas estrategias fueron seleccionadas para capturar distintos aspectos de la estructura y significado de los refranes.

3.5.1 Extracción Manual de Características

Los refranes presentan características distintivas que se pueden analizar a partir de la cantidad y tipo de elementos gramaticales presentes, como sustantivos, verbos, adjetivos y pronombres propios [58] [25] [43]. Estas características son útiles para clasificar y diferenciar los refranes de otros tipos de expresiones en los corpus. El objetivo de extraer estas características es poder identificar patrones lingüísticos específicos que faciliten la detección de refranes en textos y mejorar el desempeño del modelo de clasificación.

Estas características, denominadas Características Manuales (CM), se codificaron sistemáticamente para facilitar su referencia y análisis a lo largo de este trabajo.

3.5.1.1 Características Gramaticales

Estas características abarcan aspectos como la morfología, las sintaxis y la semántica, que permitirán analizar la estructura superficial de las oraciones y detectar patrones propios de los refranes, en la Tabla 3.7 se muestran las características que se utilizan con su respectiva descripción.

Código	Característica	Descripción
CM1	Número de sustantivos	Los sustantivos son una parte del discurso que normalmente denota una persona, un lugar, una cosa, un animal o una idea.
CM2	Número de verbos	Un verbo es un miembro de la clase sintáctica de palabras que típicamente señalan eventos y acciones,
CM3	Número de adjetivos	Palabras que normalmente modifican los sustantivos y especifican sus propiedades o atributos.
CM4	Número de adverbios	Palabras que normalmente modifican los verbos para categorías como tiempo, lugar, dirección o manera.
CM5	Número de pronombres propios	Un nombre propio es un sustantivo (o palabra de contenido nominal) que es el nombre (o parte del nombre) de un individuo, lugar u objeto específico.
CM6	Número de interjecciones	Palabras que se utiliza con mayor frecuencia como exclamación o parte de una exclamación.
CM7	Número de determinantes	Los determinantes son palabras que modifican sustantivos o frases nominales y expresan la referencia de la frase nominal en el contexto.

Tabla 3.7 Características Gramaticales

3.5.1.2 Características de Frecuencia Léxica

Otra forma de ver a los refranes considerando como parte del lenguaje figurado es el uso de palabras poco frecuentes en una oración [59], [60], en donde la frecuencia del uso de las palabras en una oración permitirá conocer si una oración cuenta con términos figurativos, como se muestra en la Tabla 3.8.

Código	Característica	Descripción
CM8	Frecuencia de palabra rara	La palabra que tenga menor frecuencia de uso. Corresponde a la frecuencia mínima encontrada entre todas las palabras que componen la oración.
CM9	Frecuencia media	El promedio de frecuencia de uso de todas las palabras contenidas en la oración
CM10	Diferencia de frecuencia	Se calcula como la diferencia absoluta entre la frecuencia de la palabra rara (mínima) y la frecuencia media

Tabla 3.8 Características de frecuencia léxica.

3.5.1.3 Características de Variación Semántica Basadas en Sinónimos

En la siguiente Tabla 3.9 se muestran otras características para captar los matices semánticos del texto, basados en la frecuencia de los sinónimos de cada palabra de una oración [5]. Son útiles para medir la riqueza léxica y la dispersión semántica.

Código	Característica	Descripción
CM11	Media de sinónimos inferiores	Se calcula el promedio de las frecuencias de uso más bajas entre los sinónimos identificados para cada palabra de la oración.
CM12	Media de sinónimos	Esta característica calcula el promedio de frecuencia de uso de todos los sinónimos correspondientes a cada palabra dentro de una frase

Tabla 3.9 Características de Variación Semántica Basadas en Sinónimos

3.5.1.4 Características Emocionales

Otra característica a considerar es la información relacionada con los sentimientos, incluyendo tanto la polaridad como las emociones [5], como se muestra en Tabla 3.10, estas características son útiles para tratar refranes, que suelen tener cargas emocionales.

Código	Característica	Descripción
CM13	Polaridad	Mide el tono emocional general del texto, indicando si el contenido es positivo, negativo o neutro. Siendo 1 positivo, -1 negativo y 0 neutro.
CM14	Introspección	Mide la presencia de emociones relacionadas con la alegría frente a la tristeza.
CM15	Temperamento	Mide la relación entre emociones asociadas a la calma frente a la ira.
CM16	Actitud	Mide la presencia de emociones relacionadas con el agrado frente al disgusto.
CM17	Sensibilidad	Refleja el nivel de entusiasmo frente al miedo.

Tabla 3.10 Características emocionales.

3.5.1.5 Características Rítmicas

Por último, se tienen las características relacionadas con las rimas, se ha considerado estas características, debido a que los refranes, al formar parte de la tradición oral, tienden a usar estructuras rítmicas para facilitar su memorización y transmisión [55], [56],[57]. En la Tabla 3.11 se muestran las características rítmicas con su respectiva descripción.

Código	Característica	Descripción
CM18	Número de palabras que riman	Cuenta el número de palabras que riman dentro de un texto. Ejemplo: “A bien obrar, bien pagar”, contiene 2 palabras que riman.
CM19	Cadencia (Patrón rítmico)	Modulación de la voz. Combinación de acentos, cortes y pausas.
CM20	Número de sílabas poéticas	

Tabla 3.11 Características rítmicas

3.5.2 Representación Mediante Embeddings

Para complementar la extracción manual de características, se considera utilizar una estrategia basada en una representación mediante *embeddings*, siendo esta una técnica utilizada en el procesamiento de lenguaje natural para capturar similitudes semánticas a partir de grandes corpus de texto.

3.5.2.1 Aplicación de FastText para la Vectorización de Refranes

Para este trabajo se ha decidido utilizar FastText [70] que es un modelo para aprender representaciones vectoriales de palabras que se distingue por su capacidad de incorporar información subléxica donde cada palabra se representa como una bolsa de n-gramas de caracteres. A cada n-grama de

caracteres se le asocia una representación vectorial; las palabras se representan como la suma de estas representaciones como se muestra en la Figura 3.6.

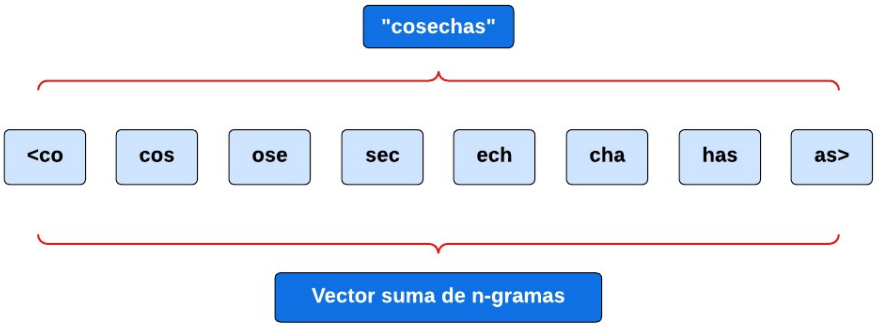

Figura 3.6 Representación de n-gramas (n=3) para la palabra "cosechas en FastText

Otra característica principal de FastText es el manejo de palabras raras y fuera de vocabulario (OOV). Al utilizar n-gramas de caracteres, FastText puede crear representaciones vectoriales para palabras que no se vieron durante el entrenamiento. Esto se logra sumando los vectores de los n-gramas de caracteres que componen la palabra. Los modelos tradicionales como Word2Vec [34] que asignan un vector único a cada palabra, como el modelo *skip-gram*, no pueden hacer esto y deben usar vectores nulos para las palabras OOV.

La información subléxica es muy útil para idiomas con muchas formas de palabras y palabras compuestas, como el alemán, el ruso y el checo. Al capturar similitudes a nivel de caracteres, FastText puede modelar mejor las relaciones entre palabras con raíces similares o afijos comunes. Por ejemplo, la palabra alemana "*Tischtennis*" (tenis de mesa) se modela mejor al considerar la similitud de caracteres con "*Tennis*".

Esta característica hace que FastText sea particularmente adecuado para el análisis de refranes, ya que estos frecuentemente contienen:

- Palabras poco comunes o arcaicas
- Variaciones dialectales regionales
- Términos específicos de la sabiduría popular
- Modismos culturalmente específicos

Al poder procesar estas peculiaridades lingüísticas a nivel de subpalabra, FastText puede mantener representaciones vectoriales coherentes incluso para términos que aparecen raramente en el corpus de entrenamiento, una ventaja significativa para el análisis computacional de expresiones populares.

3.6 Diseño de Enfoques Subsimbólicos para la Identificación de Refranes

En esta sección se explica el diseño de los enfoques que se utilizan para abordar la tarea de identificación de refranes, iniciando por la selección de los

clasificadores de aprendizaje automático y describiendo el diseño de cada uno de los enfoques subsimbólicos.

3.6.1 Selección de Clasificadores Tradicionales

Para este Trabajo de Fin de Máster se ha utilizado los siguientes clasificadores de aprendizaje automático: regresión logística, *random forest* y SVM, esta decisión se ha tomado como primer punto de partida para abordar la tarea de identificación de refranes.

3.6.1.1 Regresión Logística

Se ha considerado por su capacidad para proporcionar probabilidades interpretables y manejar eficientemente grandes conjuntos de características textuales, en lo que respecta al costo computacional este permite un entrenamiento rápido con volúmenes de datos considerables. Para la tarea de identificación de refranes, se emplea el valor umbral estándar de 0.5 [71].

3.6.1.2 Random Forest

Los refranes son expresiones lingüísticas específicas y poco frecuentes, lo que da lugar a un escenario donde se tiene escases de recursos de datos. Por esta razón se ha optado por utilizar *random forest*, debido a que este es adecuado para tareas donde se tiene pocos datos y tiene una resistencia al sobreajuste (memorizar los datos de entrenamiento) cuando se trabaja con un conjunto pequeñode datos [72].

3.6.1.3 Support Vector Machine

SVM ha sido utilizado en tareas de detección de frases figurativas, obteniendo muy buenos resultados [73], teniendo como principal ventaja su eficiencia con datos escasos, que a diferencia de las redes neuronales que requieren grandes cantidades de datos para su entrenamiento, las SVM pueden logran un gran desempeño con pocos datos de muestra y se utilizan para una clasificación binaria aunque existen estrategias para extenderlas a problemas de multiclase [74].

3.6.2 Diseño de Enfoques Basado en Clasificadores Tradicionales

El primer enfoque que se va a diseñar se basa en la regresión logística y el uso de bolsa de palabras (*Bag of Words*, BoW) para la representación textual.

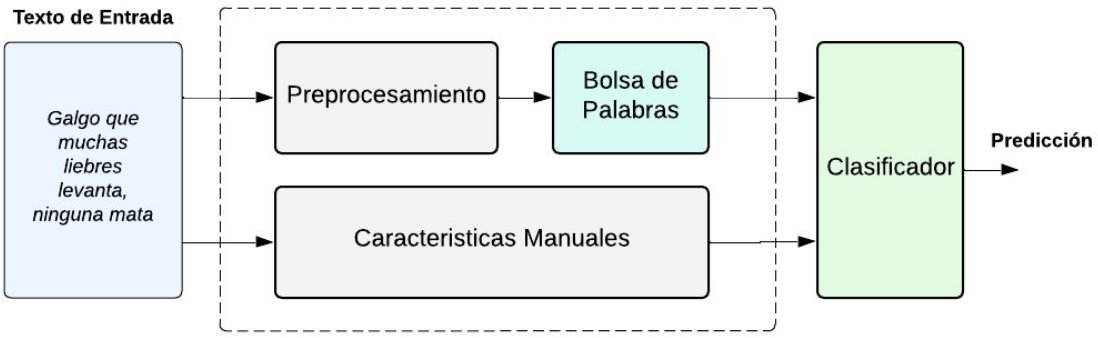

Figura 3.7 Clasificador con regresión logística.

En la Figura 3.7 se muestra la arquitectura propuesta para la identificación de refranes teniendo como última etapa un clasificador que puede ser regresión logística, *random forest* o SVM. El flujo de esta arquitectura comienza con un texto de entrada mismo que será preprocesado para tener una bolsa de palabras y extraer las características manuales del texto ingresado, para luego converger en el clasificador que finalmente generará una predicción sobre si el texto es un refrán o no.

3.6.3 Diseño de Enfoques Híbridos con Redes Convolucionales (CNN)

La arquitectura propuesta implementa un enfoque híbrido que combina características profundas extraídas mediante redes convolucionales con características extraídas manualmente. Para la representación vectorial del texto se emplea FastText, elegido por su capacidad de manejar palabras poco frecuentes y términos fuera del vocabulario, una característica valiosa en la identificación de refranes. Ya que estos suelen contener expresiones poco comunes o variaciones culturales.

El proceso de clasificación se realiza mediante algoritmos de aprendizaje automático establecidos, incluyendo regresión logística, *Random Forest* y SVM. Como se ilustra en la Figura 3.8, el sistema procesa el texto de entrada a través de dos vías paralelas: una para la extracción de características manuales y otra para el procesamiento mediante FastText y CNN. Las características resultantes de ambas vías se combinan para alimentar el clasificador final, que determina si el texto corresponde o no a un refrán.

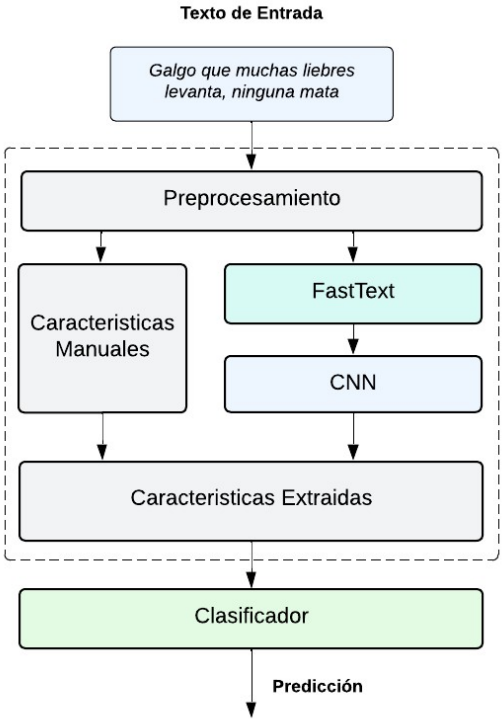

Figura 3.8 Arquitectura general para los enfoques híbridos con redes convolucionales.

Esta arquitectura híbrida aprovecha tanto el poder del aprendizaje profundo como el conocimiento incorporado en las características manuales, proporcionando un enfoque más robusto para la identificación de refranes.

3.6.4 Diseño de Enfoques Basado en Modelo preentrenado tipo encoder-decoder (FLAN-T5)

Los modelos base pre-entrenados representan un avance en el procesamiento de lenguaje natural, ofreciendo arquitecturas que han sido previamente entrenadas en grandes corpus de texto. Estos modelos aprovechan el aprendizaje por transferencia, permitiendo adaptar el conocimiento adquirido durante el pre-entrenamiento a tareas específicas mediante un proceso de ajuste fino. Esta aproximación resulta particularmente valiosa para el análisis de refranes, ya que los modelos pre-entrenados poseen un conocimiento base del lenguaje, facilitando la identificación de patrones lingüísticos complejos.

Para este trabajo se ha seleccionado el modelo FLAN-T5 [37] que es un modelo desarrollado por Google y una versión mejorada del modelo T5. Que tiene como principal idea abordar cada problema de procesamiento de texto como un problema de "texto a texto", es decir, tomando texto como entrada y produciendo un texto nuevo como salida [18], como se muestra en la Figura 3.9.

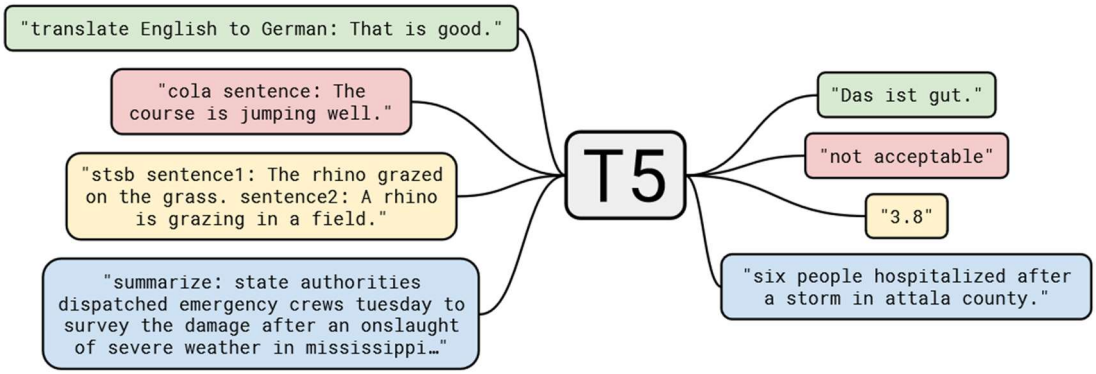

Figura 3.9 Diagrama de conversión de texto a texto del modelo FLAN-T5 [37].

FLAN-T5 aprovecha el aprendizaje por transferencia, en el que el modelo se entrena previamente en un corpus masivo de datos y luego se ajusta en tareas específicas, lo que permite mejorar el rendimiento y la eficacia. Este enfoque ha demostrado ser muy eficaz para mejorar la precisión de los modelos de PNL.

Para este Trabajo de Fin de Máster, se han seleccionado los modelos FLAN-T5-Small, FLAN-T5-Base y FLAN-T5-Large, considerando que:

- Proporcionan un rango representativo de capacidades.
- Permiten evaluar la relación entre complejidad del modelo y rendimiento.
- Mantienen requisitos computacionales manejables.
- Ofrecen un balance adecuado entre precisión y eficiencia.

Esta selección permite realizar una evaluación comparativa significativa mientras se mantiene dentro de los límites prácticos de recursos computacionales disponibles.

En Figura 3.10 se muestra la estrategia diseñada para el entrenamiento de los modelos FLAN-T5, siendo cada una de estas etapas secuenciales iniciando con la aleatorización de los datos para luego ser procesados y limpiados, seguido de esto se realiza una segmentación de los datos en subconjuntos que servirán para entrenamiento, validación y test del modelo. Las etapas finales corresponden a la incorporación de un prefijo “Clasificar” a los diferentes ejemplos y seguido se transforma el texto en representaciones numéricas procesables para el modelo FLAN-T5.

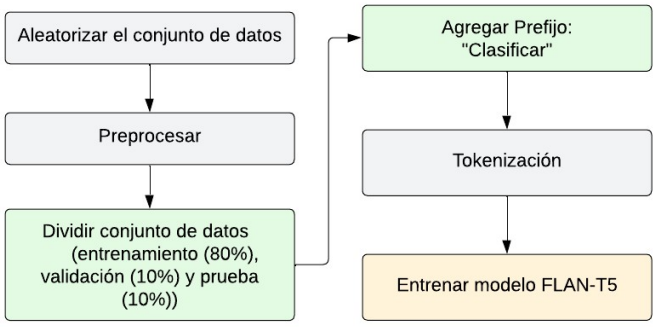

Figura 3.10 Estrategia para entrenar los modelos FLAN-T5.

3.6.5 Diseño de Enfoque Basado en Prompting con Modelo LLM (GPT)

Los Enfoques Basados en Modelos de Lenguaje de Gran Escala son utilizados en el procesamiento de lenguaje natural. Estos enfoques aprovechan la capacidad de modelos de lenguaje avanzados, como los desarrollados por OpenAI que proporciona interfaces de programación de aplicaciones (APIs) para los diferentes modelos. Estos modelos han sido entrenados con grandes corpus de texto en diferentes idiomas. Y a través de sus APIs, se pueden acceder a estos modelos y utilizarlos para tareas específicas sin necesidad de entrenar o mantener localmente.

Este enfoque ofrece varias ventajas significativas para la identificación de refranes en textos escritos en español:

- Acceso a modelos entrenados con grandes volúmenes de datos.
- Capacidad de procesamiento de lenguaje contextual sofisticado.
- Actualización continua de los modelos sin necesidad de hardware local.

En la Figura 3.11 se muestra la estrategia planteada para abordar este enfoque donde cada texto de entrada es preprocesado para eliminar caracteres especiales y normalizar los datos, para luego ser utilizado conjuntamente con un *prompt* que será enviado a la API de OpenAI, para obtener una predicción.

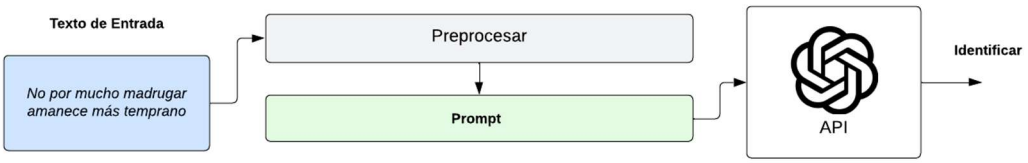

Figura 3.11 Estrategia para el uso de la API de OpenAI

En el contexto de este trabajo, se explora cómo estos servicios pueden ser aprovechados específicamente para la identificación de refranes en textos escritos en español, evaluando su efectividad y comparando su rendimiento con otros enfoques tradicionales.

3.7 Estrategia de Evaluación

La evaluación de la eficacia de las distintas técnicas subsimbólicas en la tarea de identificación de refranes, es una de las principales etapas de este trabajo, dado que el principal objetivo no se limita únicamente a alcanzar buenos resultados cuantitativos, sino también a entender el comportamiento de los diferentes enfoques frente a los diferentes escenario y tipos de entrada de texto. Por esta razón se propone una estrategia de evaluación que combina métricas estándar con un análisis cualitativo de errores.

3.7.1 Métricas de Evaluación

Para la evaluación de los diferentes enfoques se hace uso de las siguientes ecuaciones para el cálculo de la precisión, exhaustividad (*recall*), puntaje F1 (*F1-Score*) y la exactitud (*accuracy*):

3.7.1.1 Cálculo de Precisión

Se mide el porcentaje de predicciones que el modelo detectó como refranes aquellas oraciones que realmente son correctas.

$$Precisión = \frac{VP}{VP + FP} \quad (3.1)$$

3.7.1.2 Medición de Exhaustividad (Recall)

Se mide el porcentaje de los refranes reales presentes en todo el conjunto de datos que fueron correctamente identificados por el modelo.

$$Exhaustividad\ (Recall) = \frac{VP}{VP + FN} \quad (3.2)$$

3.7.1.3 Cálculo de Medida F1 (F1-Score)

Proporciona una medida balanceada que combina precisión y sensibilidad en un único valor. Esta métrica es especialmente útil para evaluar el rendimiento general del modelo.

$$Medida\ F1\ (F1 - Score) = 2\frac{Precisión\ .\ Sensibilidad}{Precisión + Sensibilidad} \quad (3.3)$$

3.7.1.4 Cálculo de Exactitud (Accuracy)

Se calcula el porcentaje total de predicciones correctas, considerando tanto la identificación acertada de refranes como de no refranes, en relación con el total de casos evaluados.

$$Exactitud\ (Accuracy) = \frac{TP\ +\ TN}{(TP + TN + FP + FN)} \quad (3.4)$$

La combinación de estas métricas proporciona una evaluación comprehensiva del rendimiento del modelo, permitiendo analizar su efectividad desde diferentes perspectivas. Esto es especialmente importante dado que la identificación de refranes presenta desafíos únicos debido a su naturaleza figurativa y variabilidad contextual.

3.7.2 Análisis Cualitativo de Errores

Como análisis adicional para evaluar los diferentes métodos utilizados en este trabajo para la identificación de refranes, se ha realizado un análisis cualitativo de errores, esto con el objetivo de profundizar en el comportamiento de los diferentes métodos de clasificación implementados y comprender las causas que llevan a un enfoque a equivocarse.

Este análisis se centra en dos tipos de errores relevantes para la tarea de identificación de refranes:

- Falsos negativos: refranes que el modelo no logra identificar como tales.
- Falsos positivos: oraciones comunes o literales que son clasificadas incorrectamente como refranes, lo que puede disminuir la precisión.

El análisis cualitativo se realizará sobre una muestra representativa de errores cometidos por cada modelo, prestando especial atención a los siguientes aspectos:

- Características lingüísticas presentes en el texto (rima, estructura simétrica, polaridad emocional, entre otros aspectos).
- Complejidad semántica o ambigüedad del contenido.
- Presencia de patrones rítmicos o métricos que pueden inducir a error.

Este análisis no solo permitirá comparar los modelos de forma funcional, sino también aportar recomendaciones sobre las fortalezas y debilidades de cada enfoque en relación con la tarea de identificación de refranes en textos en español.

4 Desarrollo

En esta sección se explica la implementación del sistema, que está compuesto por tres componentes: (1) La Interfaz de gráfica de usuario, que permitirá a los usuarios interactuar con los diferentes enfoques implementados y muestra los resultados de la identificación del texto; (2) la API que contine los diferentes enfoques implementados y permite el acceso a los mismos utilizando peticiones HTTP a través de sus *endpoints*, y finalmente, (3) el desarrollo de los diferentes enfoques. Para ello se lleva a cabo un análisis de las librerías y herramientas utilizadas para este fin, así como la integración de los diferentes métodos utilizados para la identificación de refranes con la página web.

4.1 Lenguajes y Librerías

En esta sección se detallan los lenguajes y librerías utilizados para la implementación del sistema.

4.1.1 Lenguajes y Librerías Usadas en la Interfaz gráfica de Usuario

Para el desarrollo de la interfaz de usuario se ha utilizado como lenguaje de programación TypeScript[1], que es una extensión tipada de JavaScript que aporta robustez y mantenibilidad al código. Con el objetivo de crear una interfaz gráfica intuitiva y eficiente, se han integrado las siguientes librerías:

- **React[2]:** Una biblioteca de JavaScript que soporta TypeScript de código abierto para el desarrollo de interfaces de usuario. Su arquitectura basada en componentes y su sistema de renderizado permiten crear aplicaciones web dinámicas y de buen rendimiento. React facilita la construcción de interfaces modulares y reutilizables, lo que optimiza tanto el desarrollo como el mantenimiento del código.
- **Material UI[3]:** Una librería de componentes de React que implementa las directrices de diseño de Google (*Material Design*). Esta biblioteca proporciona un conjunto completo de componentes prediseñados y personalizables como botones, campos de texto, navegación y sistemas de cuadrícula. Su implementación acelera significativamente el desarrollo.

[1] https://www.typescriptlang.org/
[2] https://react.dev/
[3] https://mui.com/material-ui/

4.1.2 Lenguajes y Librerías Usadas en la API

Para el desarrollo de la API se ha utilizado Python[4] como lenguaje principal de programación, complementado con FastApi[5], un *framework* utilizado para la construcción de APIs. La elección de FastAPI responde a múltiples necesidades del proyecto:

Rendimiento: FastAPI está construido sobre Starlette[6] y Pydantic[7], lo que permite un alto rendimiento en las operaciones y una validación eficiente de datos.

Modularidad: El *framework* facilita la implementación de *endpoints* independientes para cada método de identificación de refranes, permitiendo una arquitectura modular y escalable.

Flexibilidad: La estructura del API permite redirigir las peticiones a los diferentes métodos de identificación implementadas, proporcionando una interfaz unificada para acceder a los distintos métodos de análisis.

4.1.3 Lenguajes y Librerías Usadas para el Desarrollo de los métodos de Identificación de Refranes

En el desarrollo de los métodos de identificación de refranes, se ha empleado Python como lenguaje principal de programación, debido a su extensa colección de bibliotecas especializadas en procesamiento de lenguaje natural y aprendizaje automático. A continuación, se detallan las principales herramientas y bibliotecas utilizadas, describiendo su funcionalidad y función específica en el proyecto:

Playwright[8]: Es una librería para crear pruebas de interfaz de usuario, y puede ser utilizado para diferentes propósitos como automatización de páginas web, y *web scraping*. Se emplea para hacer un *web scraping* para obtener un corpus de refranes.

Pandas[9]: Es una herramienta de código abierto para el análisis y manipulación de datos. Se utiliza para el manejo de los diferentes corpus utilizados en este trabajo, así como el análisis de estos.

Matplotlib[10]: Es una librería para crear, animar y visualizar datos en Python. Se emplea para graficar los datos obtenidos en el proceso de análisis, entrenamiento y evaluación.

---

[4] https://www.python.org/
[5] https://fastapi.tiangolo.com/

[6] https://www.starlette.io/
[7] https://docs.pydantic.dev/
[8] https://playwright.dev/python/
[9] https://pandas.pydata.org/
[10] https://matplotlib.org/

Scikit-learn [11] : Es una biblioteca de código abierto para el aprendizaje automático que utiliza el lenguaje de programación Python. Incluye diferentes algoritmos de clasificación que se utilizan para la implementación de los métodos de identificación de refranes.

Rantanplan[12]: Es una librería de Python utilizada para obtener la medida del ritmo de los versos de un poema. Se utiliza para encontrar las silabas poéticas, ritmo y rima, características que se emplean para los métodos de identificación de refranes.

Gensim [13] : Es una biblioteca de código abierto especializada en la representación vectorial de palabras y el modelado de temas. Su principal función en este proyecto es la generación de vectores semánticos a partir de texto, empleando modelos como Word2Vec y FastText.

spaCy[14]: Es una librería para el procesamiento de lenguaje natural que ofrece herramientas como la tokenización, etiquetado gramatical y análisis de dependencias. Se emplea para extraer información y crear las características manuales que sirven para la identificación de refranes.

FastText [15] : Es una librería de código abierto que permite obtener representaciones vectoriales de textos. Es utilizado para crear la representación vectorial de las palabras de los refranes.

PyPDF2[16]: Es una librería de Python para la manipulación, extracción y análisis de documentos PDF. Se utiliza para obtener refranes de libros que se encuentran en formato PDF.

Transformers[17]: Es una librería que provee API y herramientas para descargar modelos pre-entrenados. Se emplea para usar modelos como FLAN-T5 para entrenarlo en la identificación de refranes reduciendo el costo computacional.

XML [18] : es un módulo de Python especializado en el procesamiento y manipulación de archivos en formato XML (*eXtensible Markup Language*). Para este trabajo, se utiliza específicamente para procesar los archivos anotados del corpus AnCora, permitiendo la extracción eficiente de oraciones.

Docker[19]: Es una herramienta de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software. Se emplea para tener ambientes virtuales permitiendo instalar dependencias y hacer que nuestro sistema pueda ser desplegado y compartido en otros computadores, se utiliza tanto para la API y el desarrollo de los métodos de identificación de refranes.

---

[11] https://scikit-learn.org/
[12] https://pypi.org/project/rantanplan/
[13] https://radimrehurek.com/gensim/
[14] https://spacy.io/
[15] https://fasttext.cc/
[16] https://pypdf2.readthedocs.io/
[17] https://huggingface.co/docs/transformers/en/index
[18] https://docs.python.org/3/library/xml.html
[19] https://www.docker.com/

4.2 Recolección de Refranes

Esta sección describe el proceso de recolección de refranes, fundamental para el desarrollo de los métodos de identificación de refranes, para lo cual se ha utilizado diferentes técnicas de extracción que se detallan en las secciones subsiguientes.

4.2.1 Extracción de Refranes del Centro Virtual Cervantes mediante Web Scraping

Para la recopilación de refranes en español, se utilizó como fuente el Centro Virtual Cervantes[20], que ofrece un extenso refranero organizado alfabéticamente. La extracción de datos se realizó mediante técnicas de *web scraping*, empleando Playwright como herramienta principal. Este proceso permitió recorrer sistemáticamente cada sección alfabética del refranero y obtener tanto los refranes como sus características asociadas (significado, variantes, contextos, ideas clave, entre otros). Como se puede observar en la Figura 4.1, los datos se estructuraron en formato JSON, incluyendo campos como identificador único (UUID), texto del refrán, paremia (tipo e idioma), ideas clave, significado, marcador de uso, observaciones léxicas, variantes, contextos, sinónimos y antónimos.


```
    {
        "uuid": "0a9cf312-9f22-4ae4-a182-120679c67fc6",
        "refran": "A boda ni bautizo, no vayas sin ser llamado",
        "paremia": {
            "tipo": "Refrán",
            "idioma": "Español",
            "ideas_claves": [
                "Intromisión"
            ],
            "significado": "Se recrimina a los entrometidos, especialmente cuando hay alegría y
abundancia, como sucede en los banquetes. Alude también a la cordura con la que debe vivir quien se
precia de ser honrado.",
            "marcador_de_uso": "Poco usado",
            "observaciones_lexicas": "Bautizado equivale a «bautizo».",
            "observaciones": ""
        },
        "variantes": [
            "A boda ni bautizado, no vayas sin ser llamado (Correas1627 A20; Autoridades, «boda»)",
            "A boda ni bautizo , no vayas sin ser invitado (Colombia, fuente oral)"
        ],
        "contextos": [
            "«Pues vamos a que cuando la princesa vio al caballero tan bien jateado y con tanto boato lo
reconoció y le dijo a su padre que era su salvador y que lo que quería era casarse con él, lo que
sucedió; y yo fui y vine y no me dieron nada, bien que no me echaron de ver; porque me escurrí, teniendo
presente aquello de: “A boda ni bautizado no vayas sin ser llamado”. Pues, señor, sabrá su merced cómo
después de comerse el pan de la boda se llevaban la princesa y el caballero como perro y gato, porque
como la mujer había estado tanto tiempo en poder de Lucifer, tenía un genio bragado y pintado por el lomo
que sólo el demonio la podía aguantar» (Fernán Caballero [seudónimo de Cecilia Böhl de Faber y Larrea,
quien recopiló cuentos populares, como La suegra del diablo], La suegra del diablo y otros cuentos.
Madrid: Torremozas, 2004, p. 37)."
        ],
        "sinonimos": [],
        "antonimos": []
    },
```


Figura 4.1 Ejemplo de un refrán estructurado en formato JSON extraído del Centro Virtual Cervantes.

[20] https://cvc.cervantes.es/lengua/refranero/listado.aspx

4.2.2 Extracción de Refranes de Libros

El Diccionario de Refranes de Sbarbi y Osuna [52] contiene varios refranes con su significado, pero la naturaleza del documento no tiene un formato estándar o estructurado, ocasionando que la identificación y extracción directa de sus refranes con sus correspondientes significados, resulte ser una tarea compleja. Por esta razón se ha optado por extraer dichos refranes utilizando dos pasos: El primero consiste en leer todo el PDF y extraer sus textos para luego utilizar la API de OpenAI para extraer los refranes con sus significados.

La biblioteca PyPDF2 es utilizada para la lectura y extracción del texto del documento PDF. Y como siguiente paso se utiliza la API de OpenAI, específicamente el modelo GPT-4o mini. Este modelo de lenguaje procesa el texto extraído página por página, identificando el refrán y su significado.

Todo el proceso descrito tiene un flujo sistemático que incluye:

1. Lectura y digitalización del PDF.
2. Procesamiento página por página del contenido.
3. Análisis del texto mediante la API de OpenAI
4. Extracción y estructuración de los pares refrán-significado
5. Almacenamiento en formato CSV para su posterior utilización

Esta metodología ha permitido automatizar eficientemente la extracción de información compleja. En la Figura 4.2 se puede ver la lógica que se ha utilizado.

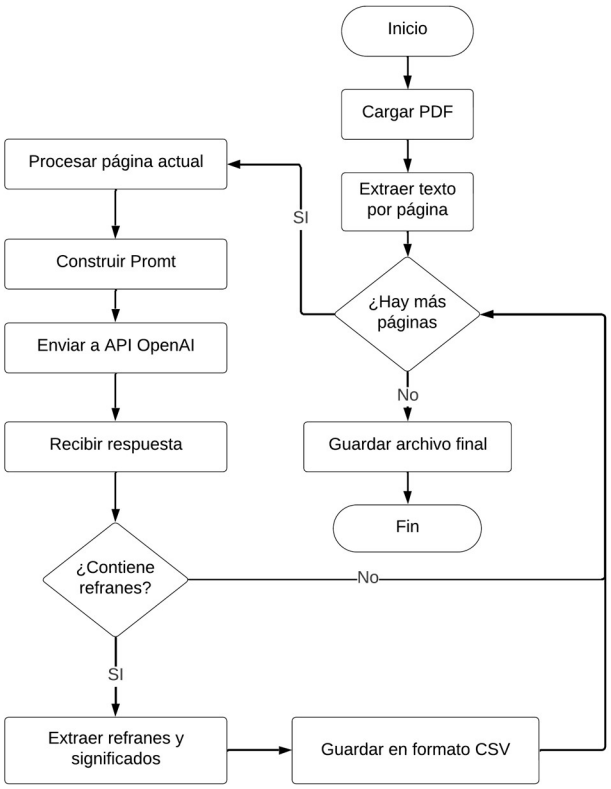

Figura 4.2 Diagrama de flujo de la lógica utilizada para extraer refranes del diccionario de Sbarbi y Asuna.

El resultado de este proceso es un archivo CSV que contiene los refranes y sus significados. Para garantizar la calidad y precisión de los datos extraídos, se realizó una revisión manual exhaustiva del archivo resultante, eliminando cualquier entrada que no correspondiera estrictamente a un refrán y su significado correspondiente.

4.3 Recolección de Oraciones Ordinarias

En esta sección se explica cómo se obtuvieron y crearon los textos que no corresponden a refranes y que se utilizan como ejemplos negativos para los diferentes métodos de identificación abordados en este trabajo.

4.3.1 Extracción de Oraciones del Corpus Ancora

Para las oraciones que no son refranes, se utilizó el corpus AnCora [54], procesado mediante la biblioteca XML[21] de Python debido a que este corpus viene estructurado en formato XML, y esta librería permite extraer sistemáticamente las oraciones anotadas.

4.3.2 Extracción de Trigramas de Refranes

Para obtener los trigramas de cada refrán se ha utilizado únicamente el corpus Cervantes, mediante la librería NLTK[22] (*Natural Language Toolkit*) y el uso de su función "*ngrams*" se obtuvieron como resultado los trigramas de cada refrán.

Por ejemplo, teniendo el refrán "**Quien come y condesa dos veces pone la mesa"**, se extraen los siguientes trigramas:

- (come condesa dos)
- (condesa dos vez)
- (dos vez poner)
- (vez poner mesa)

4.3.3 Construcción de Oraciones Literales a partir de Trigramas

En para la construcción de estas oraciones se utilizó la API de OpenAI para generar oraciones que contengan estos trigramas.

Siguiendo con el ejemplo anterior, se ilustran las oraciones obtenidas para cada trigrama:

(come condesa dos): La condesa come con sus dos amigas.

[21] https://docs.python.org/3/library/xml.html
[22] https://github.com/nltk/nltk

(condesa dos vez): La condesa salió dos veces.

(dos vez poner): Pongo la mesa dos veces porque olvidé algo.

(vez poner mesa): Puso la mesa una vez más.

4.4 Implementación de Extracción Manual de Características

En esta sección se detalla el proceso de desarrollo y la lógica utilizada para extraer las características de los refranes, mismos que se utilizan para los diferentes enfoques de identificación.

4.4.1 Extracción de Características Gramaticales

Para la extracción manual de las características de los refranes se ha empleado la información lingüística que genera Spacy por cada token, estas características a extraer se las puede ver en la Tabla 4.1.

Código	Característica
CM1	Número de sustantivos
CM2	Número de verbos
CM3	Número de adjetivos
CM4	Número de adverbios
CM5	Número de pronombres propios
CM6	Número de interjecciones
CM7	Número de determinantes

Tabla 4.1 Lista de características gramaticales.

Las características se obtuvieron utilizando el etiquetado de partes del discurso (*Part of speech tagging*). En la Tabla 4.2 se muestra un ejemplo de estas características manuales.

Texto	Texto Preprocesado	CM1	CM2	CM3	CM4	CM5	CM6	CM7
Ganar, el oro y el moro.	ganar orar morar	2	1	0	0	0	0	2
Más discurre un enamorado que cien letrados	discurrir enamorar cien letrado	2	1	0	1	0	0	1
A asno modorro, arriero loco.	asno modorrar arriero loco	0	0	0	0	0	0	0
Este kit le sigue por donde quiera que vaya	kit seguir querer ir	0	3	0	0	2	0	1
le cubrió con él la cabeza y los lloros cesaron	cubrir cabeza lloro cesar	2	2	0	0	2	0	2
A pie se llega muy rápido al hotel desde la estación	pie llegar rápido hotel estación	3	1	1	1	1	0	1

Tabla 4.2 Ejemplo de características gramaticales extraídas por Spacy.

4.4.2 Extracción de Característica de Frecuencia Léxica

Para el análisis de frecuencia de palabras, se utilizó el *Google Books Ngram Viewer*, una herramienta especializada en el análisis estadístico de frecuencias de uso de palabras y frases a través del tiempo. Esta herramienta ofrece dos métodos de acceso a los datos:

- Interfaz gráfica web, como se ilustra en la Figura 4.3, que permite realizar consultas interactivas y visualizar los resultados.
- API de consulta mediante peticiones HTTP, que devuelve los datos en formato JSON, como se muestra en la Figura 4.4. Este método es utilizado en este trabajo para automatizar del proceso de extracción de frecuencias.

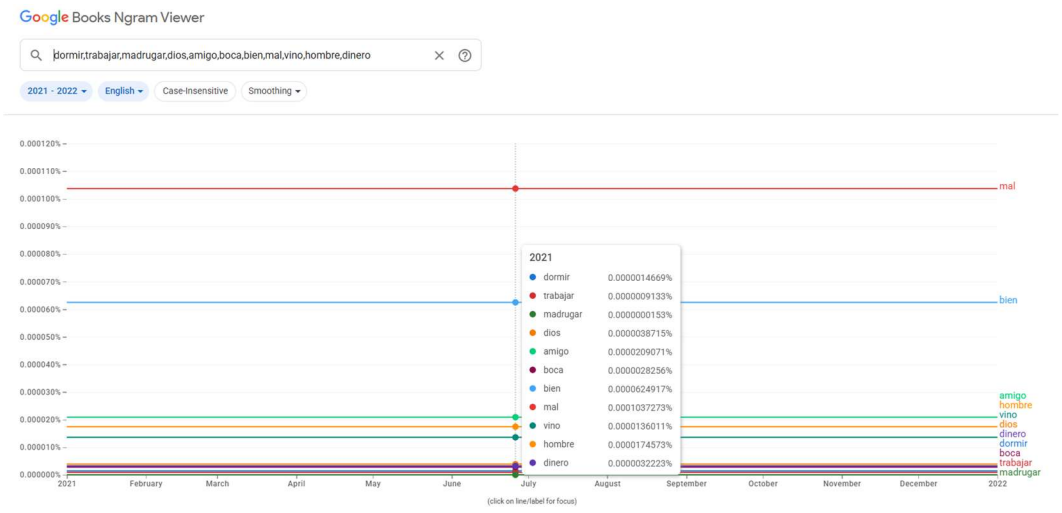

Figura 4.3 Uso interfaz gráfica de *Books Ngram Viewer*.

Figura 4.4 Respuesta en formato JSON de *Books Ngram Viewer*.

Los datos provienen del corpus de *Google Books*[23], que está disponible públicamente y contiene estadísticas de frecuencia de n-gramas extraídas de su extensa colección de libros digitalizados.

En la Tabla 4.3, se muestran las características extraídas y utilizadas para este proyecto mismos que se han descrito en la sección 3.5.1.

Código	**Característica**
CM8	Frecuencia de palabra rara
CM9	Frecuencia media
CM10	Diferencia de frecuencia

Tabla 4.3 Lista de características de frecuencia léxica.

El cálculo de la frecuencia de palabras sigue un proceso sistemático. Inicialmente, el texto se normaliza convirtiéndolo a minúsculas y eliminando elementos que no aportan información como espacios en blanco adicionales, caracteres especiales y signos de puntuación. Posteriormente, se aplica un filtrado de *stop words* (palabras vacías) y un proceso de lematización. El texto resultante se tokeniza para su análisis final. Esta lógica se lo puede observar en la Figura 4.5.

[23] https://storage.googleapis.com/books/ngrams/books/datasetsv3.html

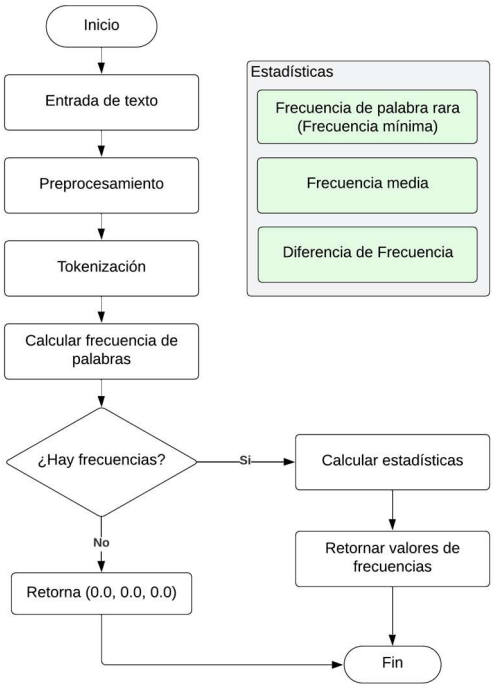

Figura 4.5 Diagrama de flujo para el cálculo de frecuencia de léxica.

En la siguiente Tabla 4.4 se muestra un ejemplo de las frecuencias obtenidas.

Texto	Texto Preprocesado	CM8	CM9	CM10
Más discurre un enamorado que cien letrados	discurrir enamorar cien letrado	4.90E-07	0.000009	0.000009
El saco vacío no puede mantenerse en pie	sacar vaciar poder mantenerse pie	2.00E-06	0.000132	0.00013
Este kit le sigue por donde quiera que vaya	kit seguir querer ir	6.76E-06	0.000182	0.000175
A asno modorro, arriero loco.	asno modorrar arriero loco	0.00E+00	0.000023	0.000023
Contra siete virtudes hay siete vicios.	siete virtud siete vicio	8.91E-06	0.000054	0.000045
le cubrió con él la cabeza y los lloros cesaron	cubrir cabeza lloro cesar	3.24E-06	0.000064	0.000061

Tabla 4.4 Ejemplo del cálculo de frecuencia de palabras.

4.4.3 Extracción de Características de Variación Semántica Basadas en Sinónimos

El cálculo de estas características se puede observar en la Figura 4.6 que utiliza el valor de la frecuencia de todos los sinónimos de cada palabra del texto y los sinónimos con frecuencia menor.

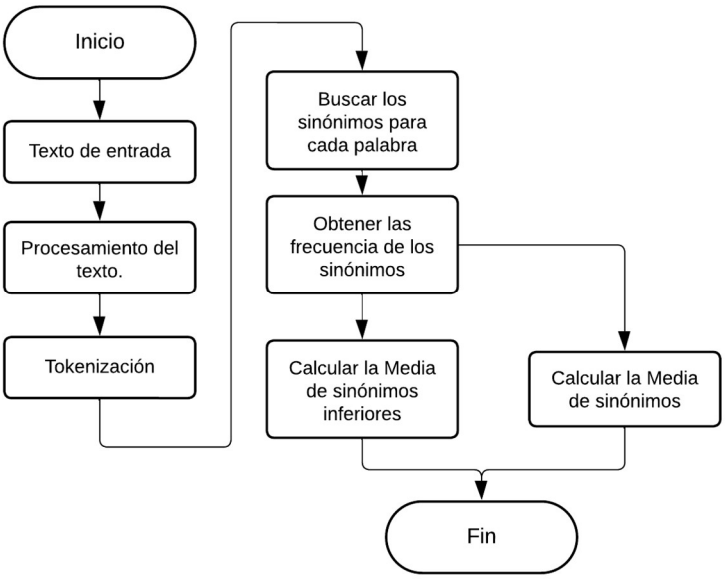

Figura 4.6 Diagrama de flujo para el cálculo de media de sinónimos inferiores y media de sinónimos.

La Figura 4.7 muestra con más detalle cómo es el proceso de obtención de los sinónimos de cada palabra.

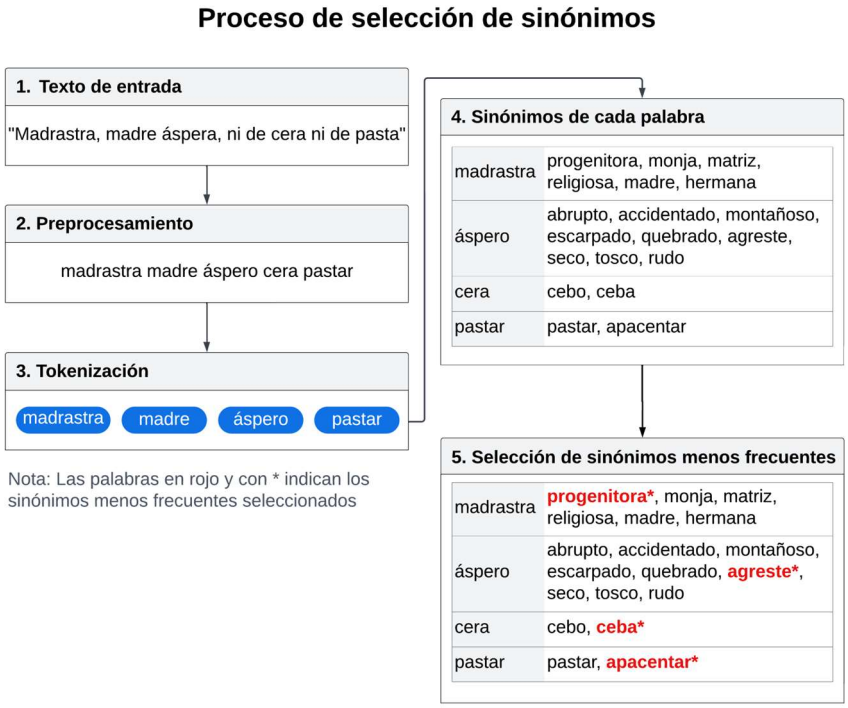

Figura 4.7 Proceso de selección de sinónimos

Siendo $S(\omega_i)$ el conjunto de sinónimos de la palabra ω_i en una oración. La frecuencia de cada sinónimo $s_{ij} \in S(\omega_i)$, donde j es el índice del sinónimo. La media de las frecuencias de los sinónimos de ω_i se calcula como:

$$mean_i = \frac{\sum_{j=1}^{|S(\omega_i)|} f(s_{ij})}{|S(\omega_i)|} \quad (4.1)$$

Pudiendo calcular la media de sinónimos del texto completo como:

$$Syn\ Mean = \frac{\sum_{i=1}^{n} mean_i}{n} \quad (4.2)$$

Donde n es el número de palabras con sinónimos relevantes en la oración.

Por otro lado, para cada palabra ω_i, se selecciona la frecuencia del sinónimo menos común:

$$lower_i = \min f(s_{ij}) \quad para \quad s_{ij} \in S(\omega_i) \quad (4.3)$$

Calculando la media de sinónimos inferiores como:

$$Syno\ Lower\ Mean = \frac{\sum_{i=1}^{n} lower_i}{n} \quad (4.4)$$

El cálculo de características basadas en sinónimos emplea las mismas herramientas utilizadas para el análisis de frecuencia de palabras. Sin embargo, en este caso, el análisis de frecuencia se aplica a los sinónimos de cada palabra que compone la oración.

En la Tabla 4.5, se muestran las características que se van a extraer, mismas que han sido mencionadas en secciones anteriores.

Código	**Característica**
CM11	Media de sinónimos inferiores
CM12	Media de sinónimos

Tabla 4.5 Lista de características de variación semántica basade en sinónimos.

Estas características, como se ejemplifica en la Tabla 4.6, se centran exclusivamente en el análisis de los sinónimos de cada palabra, proporcionando una perspectiva adicional sobre la variabilidad léxica de la oración.

Texto	Texto Preprocesado	CM11	CM12
Más discurre un enamorado que cien letrados	discurrir enamorar cien letrado	2.29E-06	0.000053
El saco vacío no puede mantenerse en pie	sacar vaciar poder mantenerse pie	1.40E-05	0.000038
Este kit le sigue por donde quiera que vaya	kit seguir querer ir	3.40E-05	0.000047
A asno modorro, arriero loco.	asno modorrar arriero loco	3.00E-07	0.00001
Contra siete virtudes hay siete vicios.	siete virtud siete vicio	4.60E-06	0.000015
le cubrió con él la cabeza y los lloros cesaron	cubrir cabeza lloro cesar	2.03E-05	0.000035

Tabla 4.6 Ejemplo de características de variación semántica basada en sinónimos.

4.4.4 Extracción de Característica Emocionales

Para el análisis de sentimientos se empleó SenticNet, una herramienta especializada que proporciona acceso a diversas características emocionales y semánticas a través de su API. La herramienta presenta ciertas limitaciones técnicas:

1. Capacidad de procesamiento:
- Límite máximo de 8.000 caracteres por consulta
- Tamaño recomendado de entrada: aproximadamente 1.000 palabras
2. Requisitos de formato:
- No admite signos de puntuación
- Restricciones en caracteres especiales
- Requiere preprocesamiento del texto para limpieza

El acceso a la API se obtiene completando un formulario en su sitio web oficial[24], y sin costo, pero teniendo acceso únicamente por un mes. Cada característica emocional requiere una llamada específica a su correspondiente *endpoint* de la API, procesando el texto previamente limpiado según los requisitos mencionados.

En la Tabla 4.7 se muestran las características de sentimientos que se utilizan en este trabajo.

[24] https://sentic.net/api/

Código	Característica
CM13	Polaridad
CM14	Introspección
CM15	Temperamento
CM16	Actitud
CM17	Sensibilidad

Tabla 4.7 Lista de características emocionales

En la Tabla 4.8 se ejemplifica el cálculo de características de sentimiento.

Texto	Texto Preprocesado	CM13	CM14	CM15	CM16	CM17
Y ahora puede usted ver que de eso ya no le queda nada	ahora poder tú ver quedo	0.00E+00	0	0	0	0
La verdad es que es una pena porque los artículos tenían mucho nivel	verdad penar artículo nivel	0.00E+00	0	0	0	0
Estaba seguro además de que yo habría sufrido más de lo que le confesaba	seguro además sufrir confesar	1.00E+00	1.95	0	-48.8	0
más discurre un hambriento que cien letrados	discurrir hambriento cien letrado	1.00E+00	0	56.7	46.6	0
ni bebas sin ver ni firmes sin leer	beber ver firme leer	1.00E+00	0	0	0	85.4
no hay olla tan fea que no halle su cobertera	olla tan feo hallar cobertero	-1.00E+00	0	0	-81.6	-96.5

Tabla 4.8 Ejemplo de características emocionales.

4.4.5 Extracción de Característica Rítmicas

En lo que respecta a las características rítmicas se ha utilizado la librería Rantanplan[25], que está especializada en el análisis métrico de poesía, así como obtener otras características como el número de silabas poéticas y su cadencia o patrón rítmico. Esta misma librería también es utilizada por la Biblioteca

[25] https://github.com/linhd-postdata/rantanplan

Virtual Cervantes[26] para el análisis de poemas, motivo por el cual se ha optado por utilizar esta herramienta. Las características que se obtienen para este trabajo se detallan en la Tabla 4.9.

Código	Característica
CM18	Número de palabras que riman
CM19	Cadencia (Patrón rítmico)
CM20	Número de sílabas poéticas

Tabla 4.9 Lista de características de rítmicas.

Rantanplan procesa conjuntos de versos para determinar sus características métricas, centrándose especialmente en los patrones de rima al final de cada verso. En la Figura 4.8, ilustra la salida de la librería, que presenta un análisis métrico donde los acentos se destacan en rojo y se identifican las sinalefas (fusión de vocales contiguas de palabras distintas en una sílaba); además, muestra el patrón rítmico o cadencia mediante signos "+" y "-", donde "+" señala las sílabas acentuadas en cada verso.

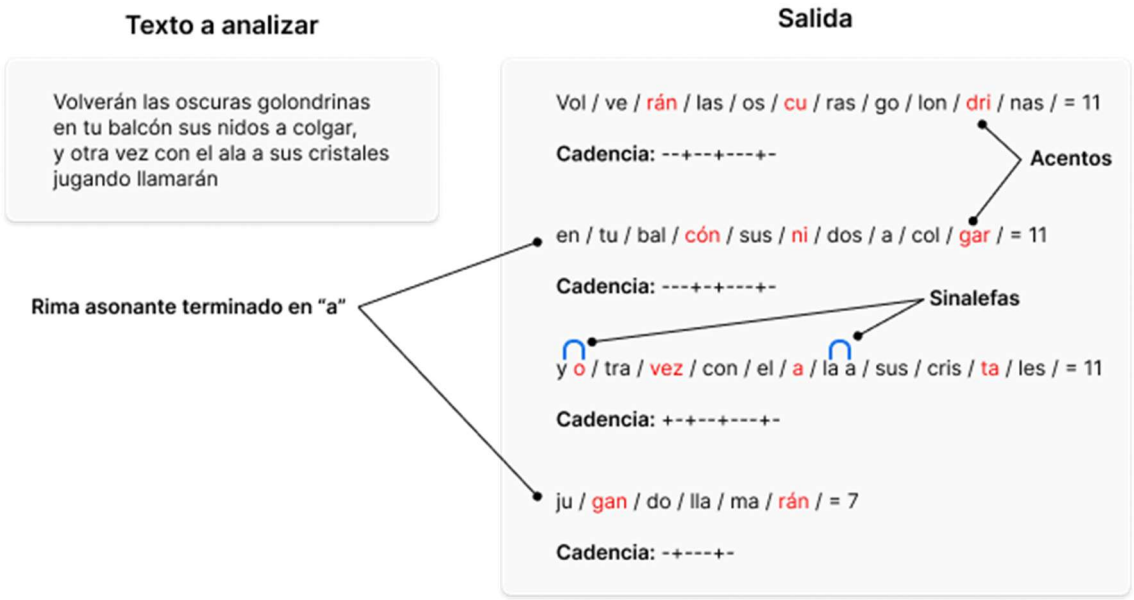

Figura 4.8 Análisis métrico y rítmico de un poema.

La salida obtenida se muestra en la Figura 4.9, siendo este un diccionario de Python con los grupos fonológicos del texto. En la figura se muestra un ejemplo del segundo y cuarto verso del ejemplo mencionado.

[26] https://data.cervantesvirtual.com/versos

```
[
    {
        "phonological_groups": [
            {"syllable": "en", "is_stressed": False, "is_word_end": True},
            {"syllable": "tu", "is_stressed": False, "is_word_end": True},
            {"syllable": "bal", "is_stressed": False},
            {"syllable": "cón", "is_stressed": True, "is_word_end": True},
            {"syllable": "sus", "is_stressed": False, "is_word_end": True},
            {"syllable": "ni", "is_stressed": True},
            {"syllable": "dos", "is_stressed": False, "is_word_end": True},
            {"syllable": "a", "is_stressed": False, "is_word_end": True},
            {"syllable": "col", "is_stressed": False},
            {"syllable": "gar", "is_stressed": True, "is_word_end": True},
        ],
        "rhythm": {
            "stress": "---+-+---+-",
            "type": "pattern",
            "length": 11,
            "length_range": {"min_length": 11, "max_length": 11},
        },
        "structure": "cuarteto_lira",
        "rhyme": "a",
        "ending": "a",
        "ending_stress": -1,
        "rhyme_type": "assonant",
        "rhyme_relaxation": True,
    },
    {
        "phonological_groups": [
            {"syllable": "ju", "is_stressed": False},
            {"syllable": "gan", "is_stressed": True},
            {"syllable": "do", "is_stressed": False, "is_word_end": True},
            {"syllable": "lla", "is_stressed": False},
            {"syllable": "ma", "is_stressed": False},
            {"syllable": "rán", "is_stressed": True, "is_word_end": True},
        ],
        "rhythm": {
            "stress": "-+---+-",
            "type": "pattern",
            "length": 7,
            "length_range": {"min_length": 7, "max_length": 7},
        },
        "structure": "cuarteto_lira",
        "rhyme": "a",
        "ending": "a",
        "ending_stress": -1,
        "rhyme_type": "assonant",
        "rhyme_relaxation": True,
    },
]
```


Figura 4.9 Resultado estructura métrica: grupos fonológicos y patrones rítmicos.

Esta librería que se utiliza necesita tener sus datos divididos en versos, y los refranes se presentan como una sola oración sin división evidente en versos, requieren un análisis especial considerando dos enfoques diferentes pero complementarios. Hay que considerar que es necesario establecer una metodología adecuada para el cálculo métrico, que permita identificar tanto las palabras que riman siendo estas las que dan el ritmo a la expresión. Por otro lado, resulta necesario definir criterios de segmentación para dividir el refrán en unidades análogas a los versos, ya sea utilizando los signos de puntuación existentes como delimitadores naturales o, en caso de su ausencia, aplicando otras técnicas de segmentación que respeten la estructura rítmica del refrán.

4.4.5.1 Lógica para el Cálculo Métrico

En esta sección se describe el proceso de extracción de características métricas del texto, considerando dos escenarios. Los refranes, como se indica en el trabajo de Anscombre [56], poseen una estructura métrica que permite su división en miembros o versos con patrones de rima. Típicamente se componen de dos versos, aunque pueden contener más. Esta división se realiza naturalmente utilizando los signos de puntuación presentes en el refrán.

Por otro lado, algunas oraciones, aunque constituyen un único verso, pueden segmentarse en unidades métricas menores. Por ejemplo, el refrán "El hábito no hace al monje" puede dividirse en un terceto:

“El hábito” / “no hace” / “el monje”

Esta segmentación permite analizar la estructura métrica incluso en refranes aparentemente simples.

Para obtener las características métricas de un texto se considera dos casos, siendo el primero el caso en que un texto no se puede dividir en versos por lo que se utiliza un análisis léxico métrico que busca las rimas de las palabras que componen el texto, mientras que en el caso de poder dividir el texto en versos se utilizaría un análisis verso métrico como se muestra en la Figura 4.10.

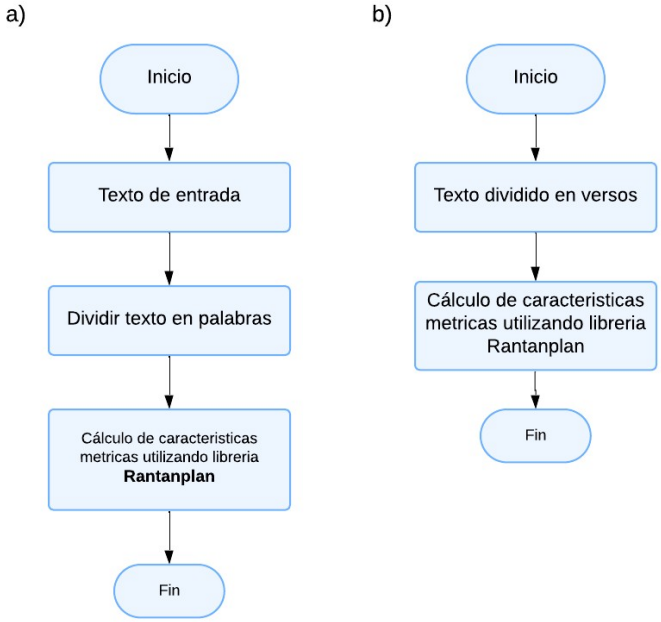

Figura 4.10 Lógica para obtener las métricas de un texto: a) análisis léxico métrico y b) análisis verso métrico.

4.4.5.2 Lógica para la Segmentación de Texto

La determinación de las características métricas del texto requiere un proceso de segmentación que sigue dos enfoques distintos.

El primer enfoque aborda los refranes que poseen una estructura natural de varios versos, identificables a través de sus signos de puntuación. Como se ilustra en la Figura 4.11, este proceso inicia con la identificación de signos de

puntuación específicos (Tabla 4.10), cuyas ubicaciones establecen los puntos para la división del texto.

Símbolo	Nombre	Ejemplo
,	Coma	Bajo la miel, está la hiel
;	Punto y coma	A lo hecho, no hay remedio; y, a lo por hacer, consejo
:	Dos puntos	Dijo el asno al mulo: 'Anda para allá, orejudo'
.	Punto	'Un día es un día', pensó el avaro. Y añadió a la olla un garbanzo

Tabla 4.10 Marcadores de puntuación utilizados para segmentar un texto en versos

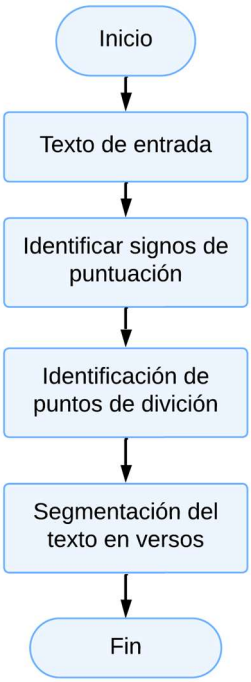

Figura 4.11 Lógica para la segmentación de texto en versos utilizando marcadores de puntuación.

El segundo enfoque se aplica a refranes que, aunque aparentemente tienen una estructura simple o unitaria, pueden segmentarse en unidades métricas menores (versos). Este método requiere un análisis detallado de la estructura sintáctica y prosódica del texto, permitiendo identificar patrones rítmicos y métricas internas que no son evidentes a primera vista.

Para estos refranes que se componen de un solo verso y no contienen signos de puntuación se ha utilizado una lógica diferente como se muestra en la Figura 4.12 que tiene dos análisis: El primero es el análisis métrico léxico, que utiliza el procesamiento de lenguaje natural (spaCy) para examinar el texto. Este análisis identifica patrones de rima y estructuras métricas que, aunque no son

inmediatamente visibles, forman parte de la composición del refrán. La identificación de estas características permite establecer puntos de división que respetan tanto el ritmo como el significado del texto.

En segundo lugar, es el análisis sintáctico, que examina la estructura gramatical del refrán para identificar patrones lingüísticos específicos. Estos incluyen elementos como conjunciones, preposiciones y negaciones, que funcionan como marcadores de división para la segmentación. Esta aproximación sintáctica complementa el análisis métrico-léxico, asegurando que las divisiones resultantes mantengan tanto la coherencia gramatical como la integridad métrica del refrán.

Los patrones sintácticos que se consideran para segmentar un texto se muestran en la Tabla 4.11.

Patrón	**Descripción**	**Punto de división**	**Ejemplo**
CONJ	Conjunción	Antes de la conjunción	Andar toda la noche / y amanecer
ADP	Preposición	Antes de la preposición	Un día / en la vida
NOUN + ADV_NEG	Sustantivo seguido de adverbio de negación	Divide después del sustantivo	El tiempo / no perdona

Tabla 4.11 Patrones sintácticos para segmentar un texto.

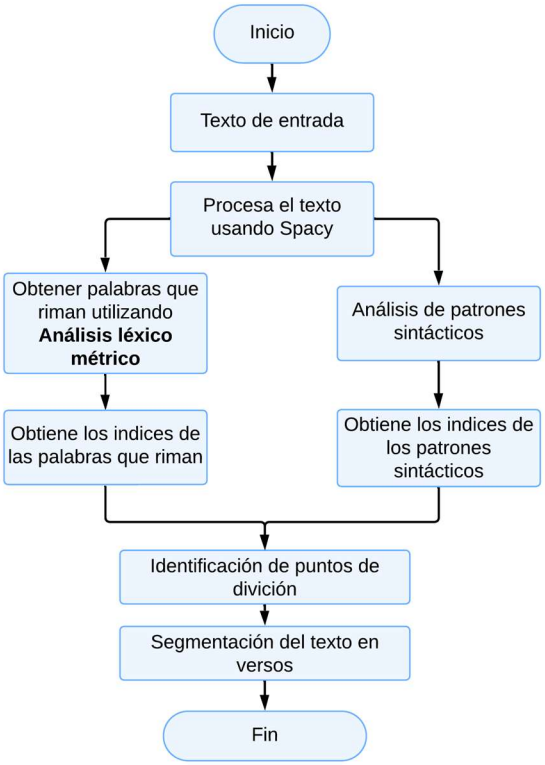

Figura 4.12 Lógica para la segmentación de texto en versos utilizando análisis metro sintáctico.

En la Tabla 4.12 se muestra un ejemplo de refranes que no pueden ser segmentados por sus marcadores de puntuación pero que utilizando el análisis metro sintáctico se pueden dividir en versos.

Texto	Segmentación
¿Adónde irá el buey que no are?	adónde irá el buey / que no are
A buen hambre no hay pan duro	a buen hambre / no hay / pan duro
A otro perro con ese hueso	a otro perro / con ese hueso
Ni a rico debas ni a pobre prometas	ni a rico debas / ni a pobre prometas
A la larga el galgo a la liebre mata	a la larga / el galgo / a la liebre mata

Tabla 4.12 Ejemplo de segmentación de texto usando análisis metro sintáctico.

4.4.5.3 Algoritmo para obtener características rítmicas de texto

Esta sección se describe la metodología utilizada para extraer características rítmicas del texto a través de un proceso de dos fases y aplicando la lógica para la segmentación de textos y el cálculo métrico descrito en la sección 4.4.5.1.

En la primera fase se implementa un análisis de división del refrán que evalúa la posibilidad de segmentar el texto en versos. Este análisis considera tanto los signos de puntuación como los patrones métrico-sintácticos que pueden revelar estructuras versales.

La segunda fase es adaptar el texto segmentado para su procesamiento con la librería Rantanplan, como se ilustra en la Figura 4.13. Esta herramienta especializada realiza un análisis rítmico, identificando patrones métricos, silábicos y acentuales que caracterizan la estructura del refrán.

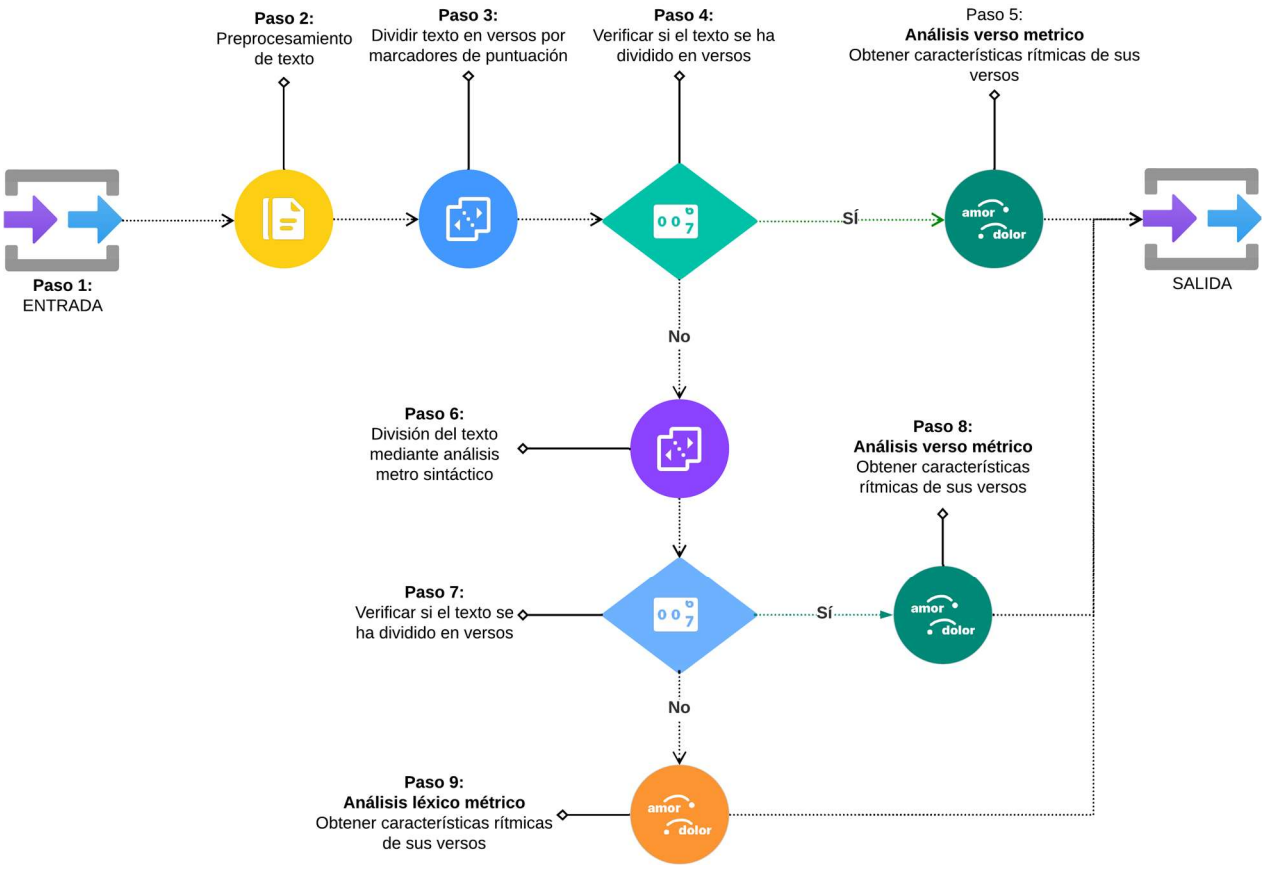

Figura 4.13 Diagrama de flujo: Procesamiento de texto para obtener características rítmicas.

Paso 1: Entrada

El proceso inicia con la recopilación del corpus, utilizando dos fuentes principales: el Corpus AnCora para oraciones generales, y una selección de refranes extraídos de la Biblioteca Virtual de Cervantes y el Diccionario de Refranes de Sbarbi. Esta combinación proporciona una base diversa para el análisis.

Paso 2: Preprocesamiento de texto

La segunda etapa consiste en el preprocesamiento del texto, donde se realiza una limpieza selectiva que preserva los elementos esenciales para el análisis métrico. Manteniendo los signos de puntuación fundamentales (comas, puntos y coma, y puntos) que servirán como marcadores de división de versos, mientras se eliminan caracteres especiales. Adicionalmente, se normaliza el texto mediante la eliminación de espacios dobles y la conversión a minúsculas, preparándolo así para su análisis posterior.

Paso 3: Dividir texto en versos por marcadores de puntuación

El proceso de división del texto en versos constituye un paso crucial para la adaptación a la librería Rantanplan. Durante esta fase, se utiliza un sistema de segmentación basado en sus signos de puntuación que funcionan como delimitadores.

Los signos de puntuación (comas, puntos y coma, y puntos) sirven como indicadores de frontera de cada verso, permitiendo una segmentación que respeta tanto la estructura sintáctica como el ritmo natural del texto. Esta división resulta esencial para que Rantanplan pueda realizar un análisis métrico preciso de cada verso. En la Tabla 4.13 se muestra un ejemplo de textos que han sido segmentados en versos utilizando sus signos de puntuación.

Texto original	**Texto dividido**
A bien obrar, bien pagar	[a bien obrar, bien pagar]
A boda ni bautizo, no vayas sin ser llamado	[a boda ni bautizo, no vayas sin ser llamado]
Al niño, mientras crece, y al enfermo, mientras adolece	[Al niño, mientras crece, y al enfermo, mientras adolece]
Donde hay celos, hay amor; donde hay viejos, hay dolor	[Donde hay celos, hay amor, donde hay viejos, hay dolor]
A buen hambre no hay pan duro	[A buen hambre no hay pan duro]

Tabla 4.13 Texto dividido en versos utilizando sus signos de puntuación.

Paso 4: Verificar si el texto se ha dividido en versos

En esta fase se implementa una verificación para determinar el método de análisis rítmico apropiado según la estructura del refrán. El proceso evalúa si el texto se ha podido segmentar usando los signos de puntuación como divisores.

Esta verificación es especialmente importante ya que existen refranes que se presentan como una única oración sin puntuación interna. En estos casos, se requiere un enfoque alternativo para la extracción de características rítmicas, utilizando otros métodos de segmentación basados en patrones sintácticos y métricos.

Paso 5: Análisis verso métrico

En el caso de refranes divisibles por puntuación, se procede a extraer las características rítmicas del texto. Este proceso analiza cada verso resultante de la segmentación para identificar sus patrones métricos, acentuales y estructurales específicos. En Tabla 4.14 la se muestra un ejemplo de estas características, donde se puede apreciar que se tiene el número de palabras que riman (CM18), el patrón rítmico representado por los signos de "+" y "-" (CM19) y finalmente el número de sílabas poéticas del texto (CM20).

Texto original	**CM18**	**CM19**	**CM20**
A bien obrar, bien pagar	2	-+-+-+-+-	9
A boda ni bautizo, no vayas sin ser llamado	0	-+---+-++--+-+-	15
Al niño, mientras crece, y al enfermo, mientras adolece	2	-+---+---+-----+-	17
Donde hay celos, hay amor; donde hay viejos, hay dolor	2	-++-+-+--++-+-+-	16

Tabla 4.14 Ejemplos de textos con sus características métricas.

Paso 6: Dividir el texto mediante análisis metro sintáctico

Para textos sin divisiones por puntuación natural, se implementa un análisis metro-sintáctico para su segmentación. Este método combina el análisis de patrones sintácticos y características métricas para identificar puntos de división apropiados, permitiendo segmentar el texto en versos que mantienen tanto la coherencia gramatical como la estructura rítmica del refrán.

Paso 7: Verifica si el texto se ha dividido en verso

En esta fase se verifica el éxito de la segmentación métrico-sintáctica del texto. Si la división ha sido exitosa, se procede con el análisis verso-métrico; en caso contrario, se aplica un análisis léxico-métrico. Esta bifurcación asegura que cada refrán reciba el tipo de análisis más apropiado según su estructura resultante.

Paso 8: Análisis verso métrico

Cuando el texto se ha logrado dividir exitosamente en versos mediante el análisis métrico-sintáctico, se procede a la obtención de características de rima utilizando el análisis verso-métrico. En Tabla 4.15 la se muestra un ejemplo de las características extraídas, siendo estas el número de palabras que riman (CM18), el patrón rítmico (CM19) y el número de silabas poéticas (CM20).

Texto original	Versos	CM18	CM19	CM20
A buen hambre no hay pan duro	[a buen hambre, no hay, pan duro]	0	-++-+-++-	9
A cada pajarillo le gusta su nidillo	[a cada pajarillo, le gusta su nidillo]	2	-+---+--+---+-	14
A cada uno lo suyo	[a cada uno, lo suyo	2	-++--+-	8
A las diez en la cama estés	[a las diez, en la cama estes]	2	--+---+-+-	10

Tabla 4.15 Ejemplo de análisis verso métrico de textos que se han segmentado por el análisis metro sintáctico.

Paso 9: Análisis léxico métrico

Cuando el texto no es divisible mediante análisis metro-sintáctico, se aplica un análisis léxico-métrico. Este enfoque examina las rimas de las palabras dentro del verso único. En la Tabla 4.16 la se muestra un ejemplo de las características extraídas, siendo estas el número de palabras que riman (CM18), el patrón rítmico (CM19) y el número de silabas poéticas (CM20).

Texto original	CM18	CM19	CM20
A cada olla su cobertera	0	-+-+----+-	10
A cada cerdo le llega su san martín	0	-+-+--+--+-+-	13
A cuerpo débil larga vida	0	-+-+-+-+-	9
Bodas hacen bodas	2	+-+-+-	6
Agua pasada no mueve molino	2	+--+-++--+-	11
Abril aguas mil	2	-+-+-+-	7

Tabla 4.16 Ejemplo de análisis verso métrico de textos que no se han segmentado por el análisis metro sintáctico.

4.5 Desarrollo de Enfoque Basado en un Clasificador de Aprendizaje Automático

En esta sección se detalla el desarrollo del primer enfoque de identificación basado en regresión logística, utilizando diversas características extraídas de los refranes. Para garantizar la robustez del modelo, se realizó un balance de datos asegurando una distribución equitativa entre refranes y oraciones que no son refranes.

4.5.1 Diseño Experimental

Se realizaron cuatro pruebas experimentales, variando tanto el tipo de preprocesamiento como las características manuales utilizadas. Para la representación textual se implementó CountVectorizer de Scikit-learn, que permite tokenizar la colección de documentos y generar un vocabulario de términos conocidos.

4.5.2 Configuración de Experimentos

El proceso experimental se estructuró variando tanto el conjunto de datos de entrenamiento como las características manuales extraídas. Los experimentos se configuraron de la siguiente manera:

Primer Experimento: Se utilizaron las características manuales CM1 a CM12 y variando con el Corpus 1 y 2. En la Figura 4.14, se muestra las características utilizadas para la identificación, así como el tipo de clasificador a utilizar.

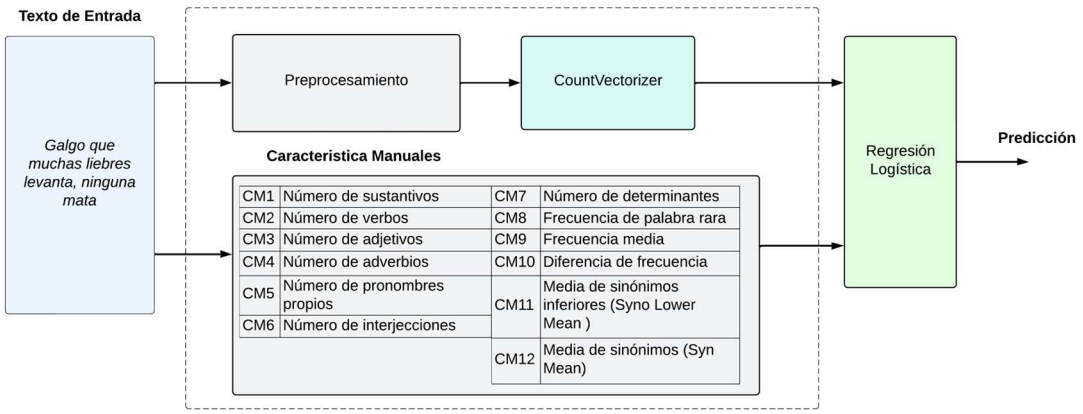

Figura 4.14 Arquitectura de modelo con regresión logística, que combina características lingüísticas manuales (CM1-CM12).

Segundo Experimento: Se incorporaron las características relacionadas con el análisis de sentimientos (CM13 a CM17) y variando con el Corpus 1 y 2. En la Figura 4.15, se muestra las características utilizadas para la identificación, así como el tipo de clasificador a utilizar.

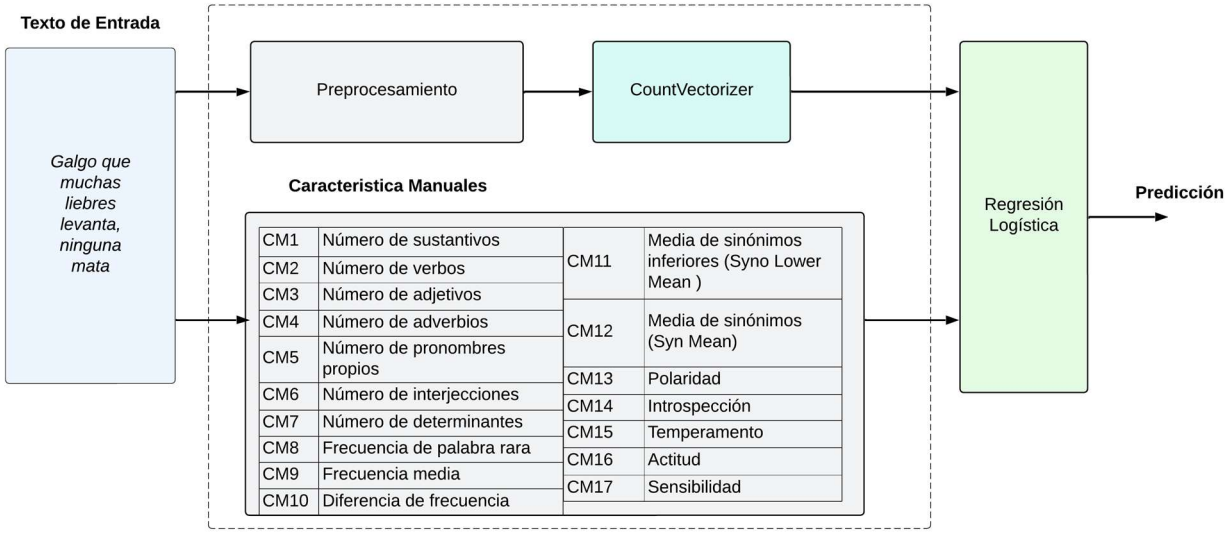

Figura 4.15 Arquitectura de modelo con regresión logística, que combina características lingüísticas manuales (CM1-CM17).

4.5.3 Arquitectura del Modelo

El sistema de clasificación se implementó mediante una estructura *pipeline* que integra dos componentes principales:

- **Preprocesamiento de Datos:** La primera etapa implementa un proceso de estandarización que normaliza las características. Esta transformación es fundamental para optimizar el rendimiento del modelo,

ya que asegura que todas las variables se encuentren en escalas comparables y contribuyan de manera equitativa al proceso de clasificación.

- **Modelo de Clasificación:** Se implementó una regresión logística como algoritmo base para la clasificación.

La optimización de hiperparámetros se realizó mediante una búsqueda aleatoria (*RandomizedSearchCV*), permitiendo una exploración eficiente del espacio de parámetros definido.

4.6 Desarrollo de Enfoques Híbridos con Redes Convolucionales

En esta sección se describe la implementación de métodos híbridos para la identificación de refranes, que combinan clasificadores tradicionales de aprendizaje automático con redes neuronales convolucionales (CNN) para obtener características profundas del texto. Este enfoque híbrido busca aprovechar tanto la capacidad de los modelos clásicos para clasificación como el potencial de las arquitecturas profundas para la extracción automática de características.

La metodología propuesta emplea tres clasificadores fundamentales del aprendizaje automático: regresión logística, *Random Forest* y SVM. Cada uno de estos clasificadores se integra con una arquitectura CNN diseñada específicamente para la extracción de características profundas del texto, permitiendo así un análisis más completo de los patrones lingüísticos presentes en los refranes.

Para asegurar un entrenamiento robusto y equilibrado, se realizó el balanceo de clases en los datos de entrenamiento. Esto es particularmente importante dado que, en los conjuntos de datos, la proporción de refranes suele ser significativamente menor que la de oraciones que no son refranes.

El proceso de preprocesamiento del texto incluye múltiples etapas: eliminación de caracteres especiales y signos de puntuación, lematización del texto y eliminación de palabras vacías (*stop words*). Esta preparación de los datos asegura que los modelos trabajen con características textuales relevantes y normalizadas, lo que mejora su capacidad de generalización y rendimiento en la tarea de identificación de refranes.

4.6.1 Arquitectura CNN para Extracción de Características

En la Figura 4.16 se muestra la arquitectura CNN implementada que está diseñada específicamente para procesar secuencias de texto y extraer características profundas mediante tres ramas paralelas de procesamiento:

1. **Estructura Multi-rama:**

Cada rama procesa n-gramas de diferentes longitudes (1-gram, 2-gram, 3-gram), estas ramas comparten una arquitectura común, pero operan de manera independiente.

2. **Componentes por Rama:**
   - Tres bloques convolucionales secuenciales.
   - Cada bloque contiene:
     - Capa convolucional 1D (64 filtros)
     - Normalización por lotes
     - Activación ReLU
     - Dropout (0,2)
   - MaxPooling global al final de cada rama.

3. **Integración de Características:**
   - Concatenación de las características extraídas por cada rama.
   - Capa densa final de 100 unidades con activación ReLU.

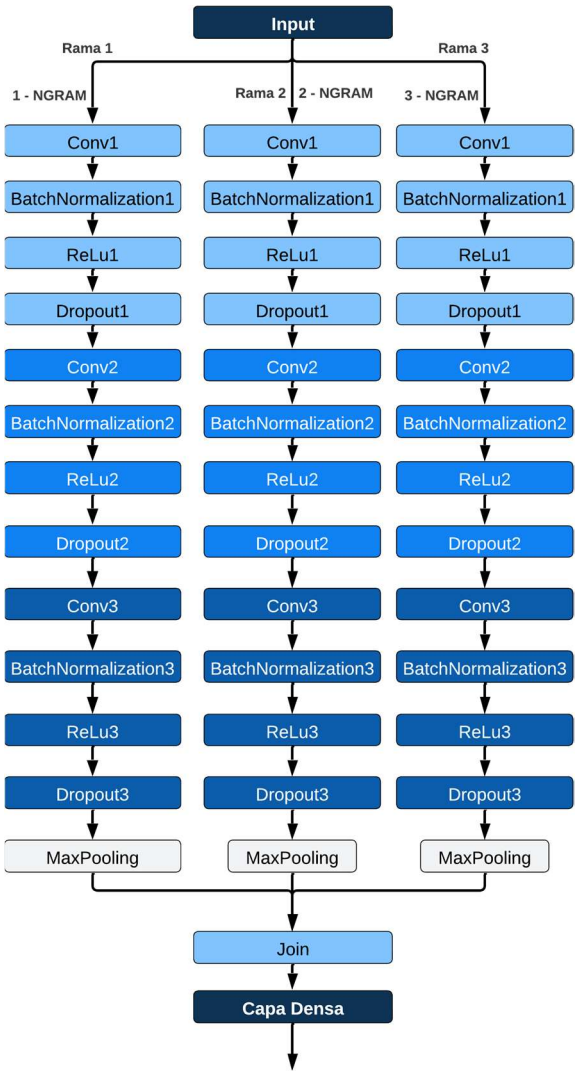

Figura 4.16 Arquitectura de la red convolucional.

4.6.2 Implementación de CNN con Regresión Logística

Este enfoque implementa una arquitectura híbrida que integra la capacidad de extracción de características profundas de una Red Neuronal Convolucional (CNN) con las propiedades de clasificación robusta de la Regresión Logística. La implementación utiliza la biblioteca Scikit-learn para la regresión logística, aprovechando su eficiencia y confiabilidad en tareas de clasificación binaria.

Para la representación vectorial del texto se ha optado por FastText, una elección fundamentada en su capacidad para manejar palabras fuera del vocabulario de entrenamiento (*OOV - Out Of Vocabulary*). Esta característica es particularmente valiosa en el análisis de refranes, donde pueden aparecer variaciones léxicas o palabras poco comunes.

Para este enfoque se han utilizado dos experimentos variando sus características manuales:

1. **Primer experimento:** utiliza el conjunto completo de características contextuales (CM1 a CM12), abarcando aspectos morfológicos y sintácticos del texto (ver Figura 4.17). En este experimento se utiliza el Corpus 5.

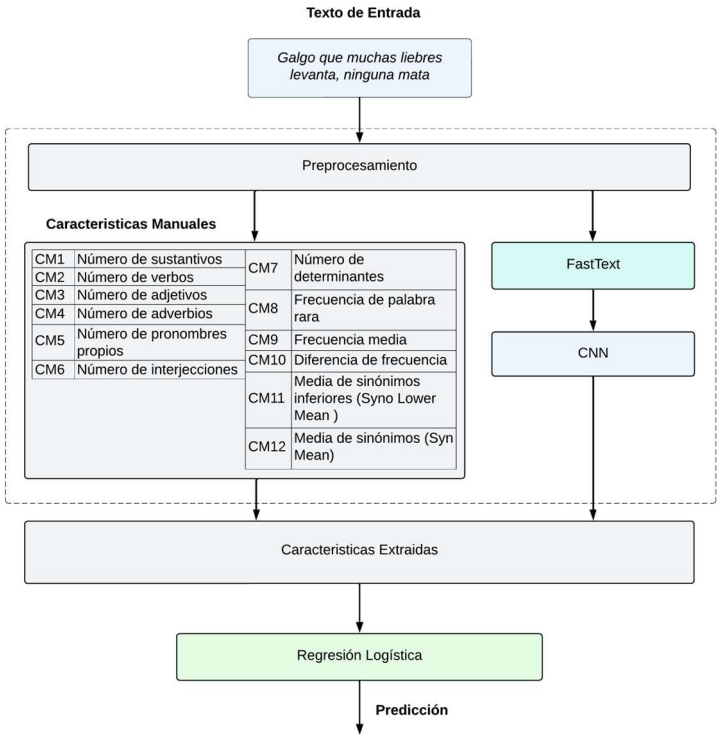

Figura 4.17 Regresión logística con CNN utilizando características morfológicos y sintácticos del texto.

2. **Segundo experimento:** se utiliza el Corpus 5 y combina dos conjuntos de características, como se ilustra en la Figura 4.18:
   - Características de frecuencia de palabras y sinónimos (CM8 a CM12).
   - Características de rima (CM18 a CM20).

Esta combinación de características profundas extraídas por la CNN y características manuales permite capturar tanto los patrones sutiles del lenguaje como las estructuras lingüísticas presentes en los refranes.

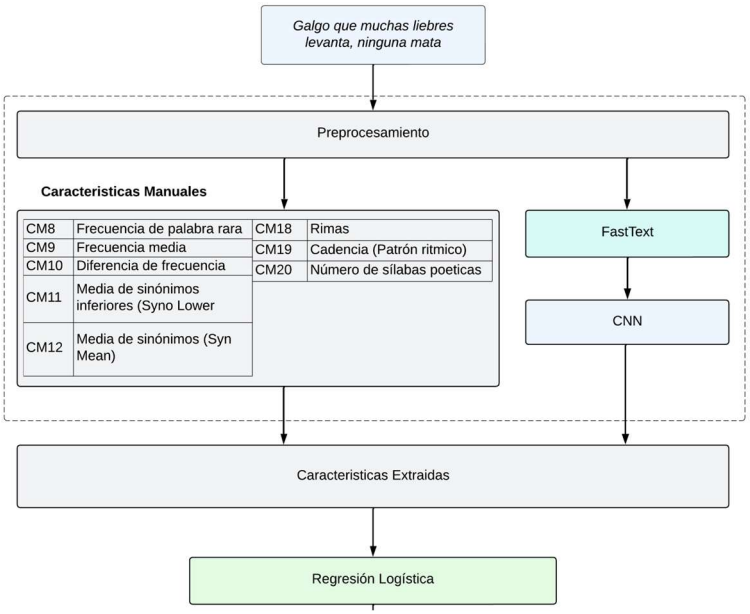

Figura 4.18 Regresión logística con CNN utilizando características de frecuencia y rima.

Proceso de Entrenamiento

El entrenamiento se realizó siguiendo estas etapas:

1. Preprocesamiento de datos:
   - División del conjunto de datos (80% entrenamiento, 10% validación)
   - Balanceo de clases.
   - Normalización de características mediante StandardScaler
2. Extracción de características:
   - Procesamiento paralelo de n-gramas a través de la CNN
   - Cálculo de características manuales
   - Concatenación de ambos conjuntos de características
3. Entrenamiento del clasificador:
   - Optimización mediante búsqueda aleatoria de hiperparámetros
   - Validación cruzada con 3 particiones
   - Evaluación continua del rendimiento

Hiperparámetros

Los principales hiperparámetros optimizados incluyen:

- Parámetro de regularización C: valores en rango [10^-4, 10^1]
- Penalización: L2
- Máximo de iteraciones: 2.000
- Ponderación de clases: balanceada
- Batch size para CNN: 64
- Tasa de dropout: 0,2

4.6.3 Implementación de CNN con Random Forest

En este enfoque se implementa un clasificador *Random Forest* (ver Figura 4.19) que combina características profundas extraídas mediante una CNN con dos conjuntos de características manuales: métricas de frecuencia de palabras (CM8-CM12) y características rítmicas (CM18-CM20). La arquitectura aprovecha la capacidad del *Random Forest* para manejar relaciones no lineales entre características y su robustez ante el sobreajuste, mientras mantiene la interpretabilidad de las características manuales seleccionadas. Este diseño permite capturar tanto los patrones léxicos y semánticos a través de las frecuencias de palabras, como los elementos estructurales del refrán mediante las características rítmicas.

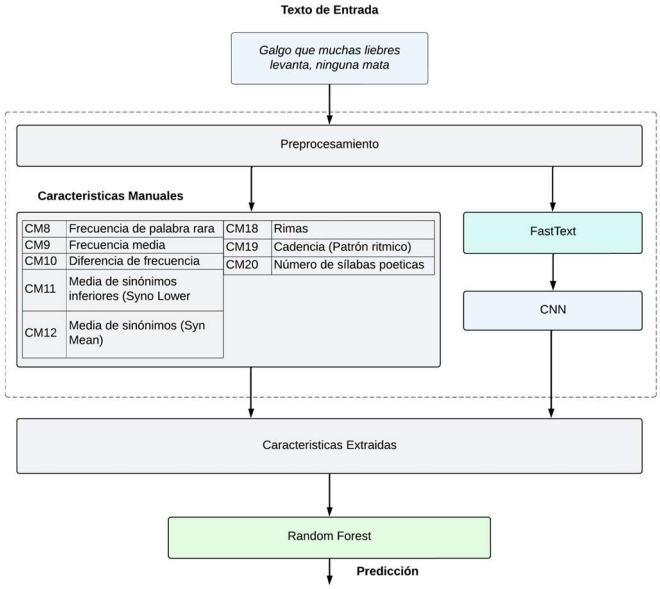

Figura 4.19 *Random Forest* con CNN utilizando características de frecuencia y rima.

Proceso de Entrenamiento

El entrenamiento se estructuró en las siguientes fases:

1. Preparación de datos:
   - División 80% entrenamiento, 10% validación
   - Balanceo de clases del Corpus 5 (15.000 instancias por categoría)
   - Aleatorización múltiple de datos
2. Extracción de características:
   - Procesamiento mediante CNN
   - Cálculo de características manuales
   - Integración de características
3. Optimización del clasificador:
   - Búsqueda aleatoria de hiperparámetros
   - Validación cruzada con 3 particiones
   - Evaluación de métricas de rendimiento

Hiperparámetros

Los hiperparámetros principales optimizados incluyen:

- Número de árboles: rango [100, 500]
- Profundidad máxima: rango [10, 50]
- Muestras mínimas para división: [2, 5, 10]
- Muestras mínimas por hoja: [1, 2, 4]
- Ponderación de clases: balanceada
- Batch size CNN: 64
- Tasa de dropout: 0,2

4.6.4 Implementación de CNN con SVM

Este enfoque implementa una SVM que integra características profundas extraídas mediante una CNN con dos conjuntos de características manuales: métricas de frecuencia de palabras y sinónimos (CM8-CM12) y características rítmicas (CM18-CM20), como se muestra en la Figura 4.20.

La arquitectura aprovecha la capacidad del SVM para crear hiperplanos óptimos de separación en espacios de alta dimensionalidad, mientras mantiene la interpretabilidad de las características manuales seleccionadas. Este diseño permite capturar tanto los patrones léxicos y semánticos a través de las frecuencias de palabras y sinónimos, como los elementos estructurales del refrán mediante las características rítmicas.

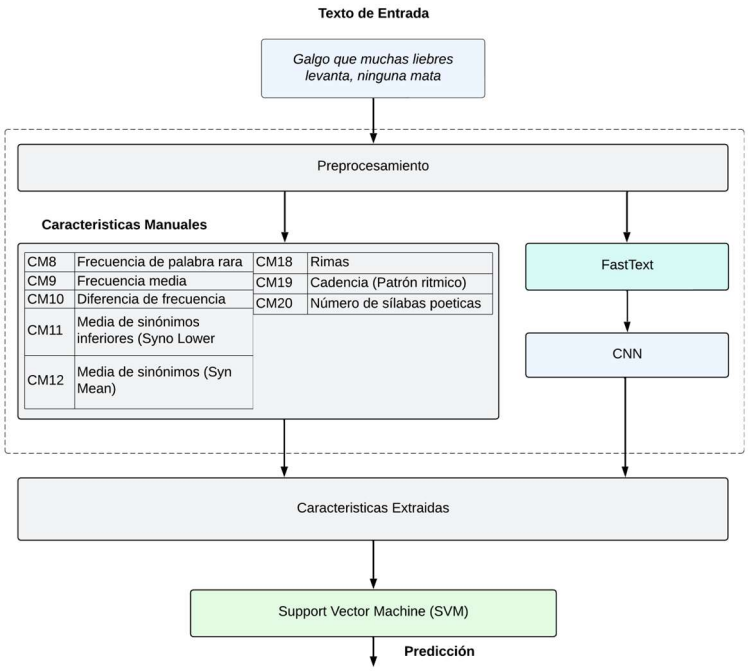

Figura 4.20 *Support Vector Machine* con CNN utilizando características de frecuencia y rima.

Proceso de Entrenamiento

El entrenamiento se organizó en las siguientes etapas:

1. Preparación de datos:
   - División 80% entrenamiento, 10% validación
   - Balanceo de clases del Corpus 5 (15.000 instancias por categoría)
   - Estandarización de características mediante StandardScaler
2. Extracción y procesamiento de características:
   - Extracción mediante CNN
   - Cálculo de características manuales
   - Normalización y concatenación de características
3. Optimización del clasificador SVM:
   - Búsqueda aleatoria de hiperparámetros
   - Validación cruzada con 3 particiones
   - Monitoreo de métricas de rendimiento

Hiperparámetros

Los principales hiperparámetros optimizados incluyen:

- Parámetro de regularización C: valores en rango [10^-3, 10^3]
- Tipos de kernel: RBF y lineal
- Parámetro gamma: 'scale', 'auto' y valores en rango [10^-3, 10^3]
- Máximo de iteraciones: 5.000
- Ponderación de clases: balanceada
- Batch size CNN: 64
- Probabilidad habilitada: True
- Tasa de dropout: 0,2

4.7 Implementación Enfoques Basados en Modelos Base Pre-entrenados FLAN-T5

Este enfoque implementa una arquitectura basada en el modelo FLAN-T5 de Google [37], una versión mejorada del T5 preentrenada en múltiples tareas. La implementación explora tres variantes del modelo: FLAN-T5-small, FLAN-T5-base y FLAN-T5-large, cada una adaptada específicamente para la tarea de identificación de refranes mediante técnicas de fine-tuning.

El desarrollo utiliza la biblioteca Transformers de Hugging Face[27], configurando cada variante del modelo para procesar el texto de entrada con el prefijo "clasificar: " seguido del refrán o texto a analizar. La arquitectura mantiene el enfoque texto-a-texto característico de T5, pero adaptando la salida para producir una clasificación binaria que distingue entre refranes y no refranes siendo 1 la salida al identificar el texto de entrada como refrán y 0 en caso contrario.

[27] https://huggingface.co/docs/transformers/en/model_doc/flan-t5

Cada variante del modelo se implementa con diferentes configuraciones y optimizaciones:

- FLAN-T5-small[28] se utiliza como base para experimentar con diversas técnicas de procesamiento, incluyendo texto plano, trigramas como ejemplos negativos y características de los refranes como la rima.
- FLAN-T5-base[29] se implementa con modificaciones en el formato de salida y ajustes en los parámetros de entrenamiento.
- FLAN-T5-large[30] incorpora adaptaciones específicas para el manejo eficiente de recursos computacionales y una salida textual más descriptiva ("es refrán"/"no es refrán").

4.7.1 Preparación del Entorno de Desarrollo

Para el desarrollo de este enfoque se comienza con la configuración del entorno de trabajo, estableciendo las dependencias necesarias para la implementación y entrenamiento del modelo. La base del entorno se construye utilizando una serie de bibliotecas especializadas en procesamiento de lenguaje natural y aprendizaje profundo:

- **Transformers[31]:** proporciona la implementación base del modelo FLAN-T5 y sus componentes
- **Datasets[32]:** facilita el manejo y procesamiento eficiente de los datos de entrenamiento
- **Spacy[33]:** ofrece herramientas para el procesamiento de texto
- **Tensorboard[34]:** facilita el seguimiento y visualización del entrenamiento

Para el manejo de datos y manipulación de tensores, se incorporan las siguientes bibliotecas:

- **NumPy[35]:** para operaciones numéricas eficientes
- **Pandas[36]:** para el procesamiento y manipulación de datos tabulares
- **PyTorch[37]:** como *backend* para el modelo de *deep learning*
- **Scikit-learn[38]:** para utilidades de preprocesamiento y métricas de evaluación

---

[28] https://huggingface.co/google/flan-t5-small
[29] https://huggingface.co/google/flan-t5-base
[30] https://huggingface.co/google/flan-t5-large
[31] https://huggingface.co/docs/transformers/en/index
[32] https://huggingface.co/docs/datasets/en/index
[33] https://spacy.io/
[34] https://www.tensorflow.org/tensorboard
[35] https://numpy.org/
[36] https://pandas.pydata.org/
[37] https://pytorch.org/
[38] https://scikit-learn.org/stable/

Esta configuración proporciona un entorno robusto y flexible para el desarrollo, entrenamiento y evaluación del modelo de identificación de refranes

Adicional a esto se hace uso de Hugging Face Hub [39] permitiendo el acceso al repositorio de modelos y facilitando el almacenamiento de los *checkpoints* durante el entrenamiento. La configuración se realiza mediante variables de entorno, utilizando python-dotenv[40] para la gestión segura de credenciales.

4.7.2 Implementación con Variantes de FLAN-T5

En la implementación de este enfoque se exploran tres variantes del modelo FLAN-T5, cada una con diferentes capacidades y configuraciones adaptadas a la tarea de identificación de refranes. Se utiliza el Corpus 2 como base (salvo que se especifique otro corpus), el cual se divide en tres conjuntos: entrenamiento, validación y prueba, con una distribución de 80%, 10% y 10% respectivamente.

FLAN-T5-small: representa la versión más ligera del modelo, utilizada como base para experimentar con diferentes técnicas de procesamiento. Se implementaron seis configuraciones principales:

1. Se configura la de entrada de texto utiliza un formato simple y directo. El texto se ingresa sin prefijos ni modificadores adicionales. Por ejemplo, para la frase “Nada nuevo bajo el sol”, se asigna una etiqueta binaria donde 0 indica que no es un refrán y 1 indica que sí lo es.

```
1 {'text': 'nada nuevo bajo el sol', 'label': '1'}
```

2. Para este experimento se agrega el prefijo "clasificar:" al texto que se va a identificar.

```
1 {'text': 'clasificar: nada nuevo bajo el sol', 'label': '1'}
```

3. En este experimento se mantiene el prefijo utilizado, pero se enriquece el conjunto de datos de entrenamiento mediante la incorporación de trigramas como ejemplos negativos (Corpus 3) para mejorar la capacidad de discriminación del modelo.
4. El experimento se utiliza el Corpus 4 que contiene oraciones construidas a partir de los trigramas más frecuentes en los refranes.
5. Se realiza un experimento adicional utilizando el Corpus 5, que combina los trigramas y las oraciones derivadas de estos.
6. Finalmente, como último experimento para esta versión del modelo se utiliza el Corpus 5 pero agregando características de ritmo, número de sílabas y el número de rimas.

[39] https://huggingface.co/docs/hub/en/index
[40] https://github.com/theskumar/python-dotenv

```
{'text': """clasificar: a cada pajarillo le gusta su nidillo.\n
            con las siguientes características:\n
            ritmo: -+---+--+---+-\n
            número de sílabas: 14\n
            rimas: 2\n
        """,
 'label': '1'
 }
```

FLAN-T5-base: se implementó como una versión intermedia, balanceando capacidad y eficiencia. Las principales adaptaciones incluyeron:

1. Se configura la de entrada de texto utiliza un formato simple y directo. El texto se ingresa sin prefijos ni modificadores adicionales.
2. Al igual que en el modelo FLAN-T5-small se utiliza el prefijo “Clasificar:” al texto que se va a identificar.

FLAN-T5-large: siendo la variante más robusta, requirió adaptaciones específicas para su implementación:

1. En este experimento se utiliza el Corpus 5, y se utiliza el prefijo “Clasificar:” al texto que se va a identificar.

```
{
  'text': clasificar: obra empezada, medio acabada’,
  'label': '1'
}
```

Cada variante del modelo se inicializó utilizando AutoModelForSeq2SeqLM y AutoTokenizer de la biblioteca Transformers, con configuraciones específicas para el tokenizador según los requerimientos de cada versión. Los modelos se configuraron para procesar textos de entrada con una longitud máxima de 512 tokens y generar predicciones con una longitud máxima de 150 tokens.

4.7.3 Procesamiento de Datos para implementarlos con FLAN-T5

El procesamiento implementa una serie de transformaciones y preparaciones de datos necesarias para el entrenamiento del modelo. El proceso comienza con la carga del conjunto de datos mediante Pandas, específicamente de dos campos: el texto que contiene la frase a identificar y la etiqueta que determina si esta corresponde o no a un refrán.

Como paso siguiente se preprocesan los datos siendo: la normalización del texto, el formateo mediante la adición o no de prefijos según el experimento, y la conversión de etiquetas a formato texto. Estas transformaciones aseguran la consistencia del formato de entrada y preparan los datos para su posterior procesamiento.

La fase final del procesamiento implica la tokenización utilizando el tokenizador de FLAN-T5. Este paso incluye el cálculo dinámico de la longitud máxima de los

tokens basado en los textos de entrada, el cálculo de la máxima longitud para las etiquetas, y la implementación de *padding* y truncamiento para manejar secuencias de diferentes longitudes. Este sistema base proporciona los datos procesados en el formato requerido por el modelo FLAN-T5, preparándolos adecuadamente para la fase de entrenamiento.

4.7.4 Proceso de Entrenamiento de Modelos FLAN-T5

El proceso de entrenamiento se implementa de manera específica para cada variante del modelo FLAN-T5, adaptando los hiperparámetros y configuraciones según las capacidades y requerimientos de cada versión.

FLAN-T5-small: el entrenamiento se configura con un *batch size* de 16, permitiendo un procesamiento eficiente de los datos. Se establece un *learning rate* de 3e-4 y el entrenamiento se ejecuta durante 2 *epochs*. Esta configuración aprovecha la naturaleza ligera del modelo, permitiendo un entrenamiento rápido.

FLAN-T5-base: requiere ajustes en sus parámetros de entrenamiento debido a su mayor tamaño. Se reduce el *batch size* a 8 para manejar el incremento en el uso de memoria. El learning rate se mantiene en 3e-4, pero el número de *epochs* varía entre 2 y 5, permitiendo una exploración más profunda del espacio de parámetros. Esta configuración busca un balance entre la capacidad de aprendizaje del modelo y los recursos computacionales disponibles.

FLAN-T5-large: siendo la variante más robusta, se implementan ajustes significativos en la configuración del entrenamiento. El *batch size* se reduce a 2 para manejar las limitaciones de memoria GPU, manteniendo el *learning rate* en 3e-4. El entrenamiento se extiende hasta 10 *epochs*.

Todas las variantes comparten características comunes en su configuración de entrenamiento, incluyendo la utilización de Seq2SeqTrainer de Hugging Face, la implementación de evaluación por *epoch*, y el almacenamiento de *checkpoints*. El progreso del entrenamiento se monitorea mediante Tensorboard, y los modelos se guardan automáticamente en Hugging Face Hub para su posterior utilización.

4.8 Implementación de Enfoque basado en Prompting con GPT para la Identificación de Refranes

En esta sección se detalla el proceso de implementación de la identificación de refranes utilizando el modelo GPT-4o mini a través de la API de OpenAI. El sistema se desarrolla mediante *prompting*. Este enfoque se diseña para evaluar la capacidad del modelo para identificar características particulares de los refranes que permitan identificar si un texto es un refrán o una oración ordinaria. Para la evaluación de este enfoque se utiliza el conjunto de prueba del Corpus 2, que representa el 10% del conjunto total de datos.

4.9 Preparación del Entorno de Desarrollo

Para el desarrollo de este enfoque se comienza con la instalación de las dependencias necesarias, siendo estas la librería de OpenIA para la interacción con la API de GPT.

Un componente crucial es la configuración de la autenticación mediante la API Key de OpenAI, que permite el acceso seguro a los modelos. La inicialización del cliente se realiza mediante la clase OpenAI:

```
from openai import OpenAI
client = OpenAI(
    api_key="your-api-key"
)
```

4.10 Implementación del Sistema de Identificación

Con la preparación del entorno de desarrollo se procede con la implementación del sistema que está compuesto por cuatro partes principales como se muestra en la Figura 4.21:

1. **Inicialización con cliente de OpenAI:** esta recibe la API Key para la autentificación, así como el nombre del modelo que se va a utilizar.
2. **Preprocesamiento de texto:** Se realiza una limpieza del texto removiendo caracteres especiales, pero manteniendo los signos de puntuación para mantener el significado.
3. **Creación del prompt:** para este paso se ha utilizado *zero shot* y se espera como respuesta las etiquetas (refrán/no refrán).

```
Clasifica el siguiente texto como: ["refrán", "no refrán"]
Texto: <oración>
Responde únicamente con la etiqueta correspondiente.
```

4. **Procesamiento de la respuesta:** se convierte los datos de salida de la API a formato textual obteniendo las etiquetas (refrán/ no refrán).

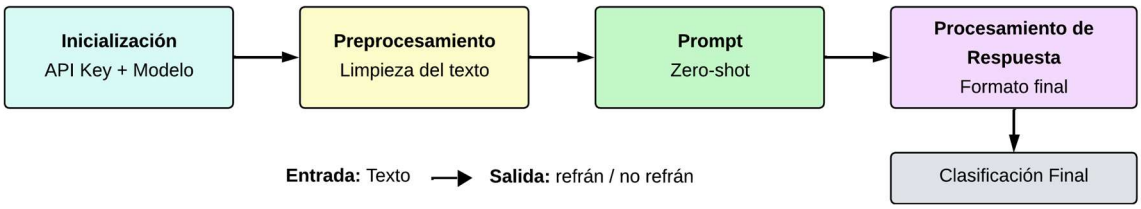

Figura 4.21 Flujo de clasificación con GPT.

Cada componente cumple una función específica en el pipeline de clasificación, desde la preparación de los datos hasta la obtención del resultado final.

4.11 Desarrollo de Página Web

En esta sección se detalla el proceso de desarrollo de la página web, misma que permite interactuar con los diferentes métodos desarrollados en este trabajo con el objetivo de comparar sus resultados.

La página web incluye una interfaz gráfica donde el usuario puede ingresar un texto para identificar si corresponde a un refrán o no. En caso de que el texto sea reconocido como un refrán, el sistema consultará una base de datos para proporcionar su significado y posibles variaciones. La propuesta de interfaz gráfica se ilustra en la Figura 4.22.

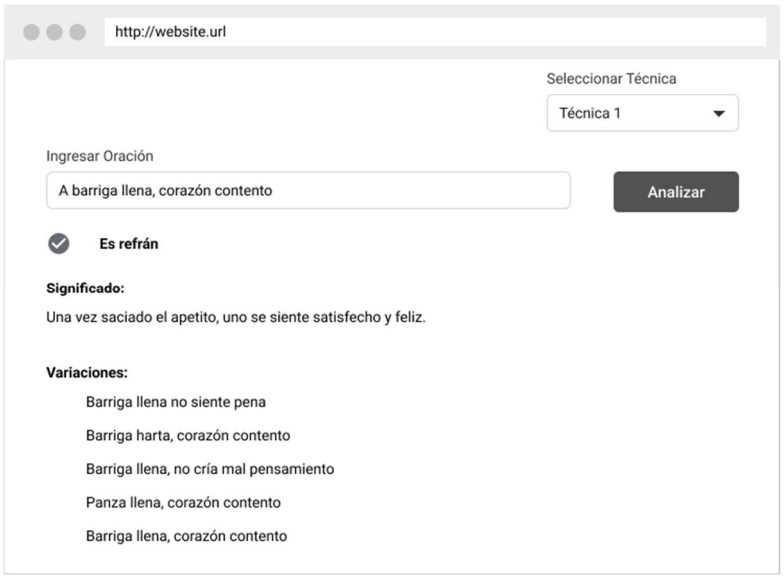

Figura 4.22 Propuesta de interfaz gráfica para la identificación de refranes.

La lógica propuesta para esta página web contempla un campo de entrada para la oración que se desea analizar. El texto ingresado será evaluado por el modelo seleccionado para determinar si es un refrán. Si la evaluación es positiva, se recuperarán los datos correspondientes desde la base de datos. Además, se integrará una serie de servicios existentes de adaptación en base a la Metodología de Lectura Fácil. Estos servicios se utilizan en la aplicación FACILE [77]. La explicación del refrán se adaptará a un lenguaje accesible usando dichos servicios. La lógica completa de esta funcionalidad se detalla en la Figura 4.23.

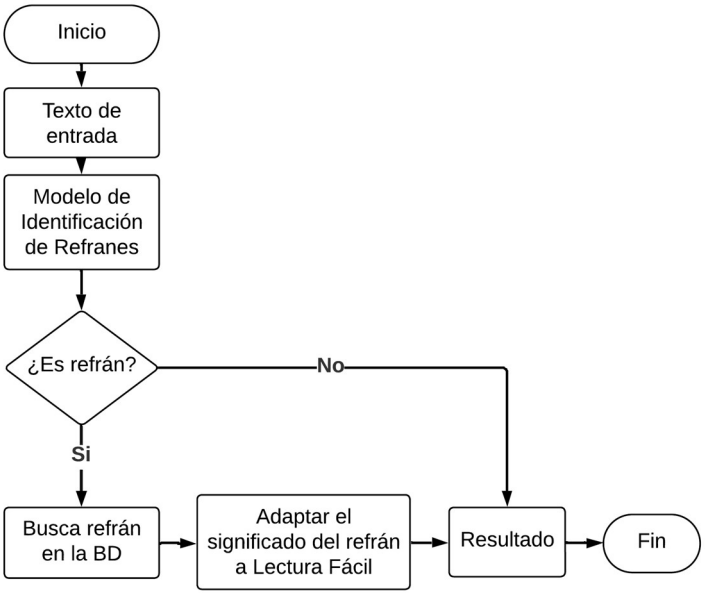

Figura 4.23 Lógica propuesta para la página web de identificación de refranes.

Para la implementación se ha estructurado en tres componentes principales que trabajan de manera coordinada para proporcionar una experiencia completa al usuario, teniendo los siguientes componentes:

Buscador semántico: Es el primer componente y este contiene una base de datos de los refranes que se han recopilado en las secciones anteriores. Este buscador permitiendo hacer una consulta específica de refranes y proporciona como salida el significado del refrán, sus variantes, antónimos y sinónimos. La importancia de este componente radica en su capacidad para proporcionar el significado completo del refrán cuando existe en la base de datos. Adicional a la respuesta que da el buscador semántico se ha utilizado el servicio de lectura fácil para adaptar el significado del refrán.

API: Como segundo componente se ha desarrollado una API que contiene los diferentes enfoques abordados en este Trabajo de Fin de Máster. La API se ha diseñado con diferentes *endpoints*, donde cada uno corresponde a un enfoque específico para la identificación de refranes. Esta estructura permite evaluar y probar cada implementación de manera independiente, facilitando la comparación de los resultados entre los diferentes métodos.

Interfaz Gráfica de Usuario: Por último, tenemos la interfaz gráfica de usuario o *GUI* por sus siglas en ingles. Esta interfaz permite interactuar directamente con la API y el buscador semántico. Su funcionamiento está basado en el análisis del texto ingresado, para determinar si dicho texto es un refrán, y en el caso afirmativo, se obtiene de la base de datos información adicional del refrán ingresado.

Esta arquitectura modular permite una experiencia de usuario fluida y completa, combinando las capacidades de clasificación de los diferentes enfoques con el acceso a una base de conocimiento detallada sobre refranes.

4.11.1 Arquitectura y Tecnologías

En esta sección se detalla la arquitectura y tecnologías utilizadas para el desarrollo de la página web. Para esto se ha utilizado dos repositorios diferentes uno para la interfaz de usuario y otro que alberga la API con el buscador semántico.

Como se ilustra en la Figura 4.24, la interfaz de usuario se ha desarrollado utilizado React[41] una librería de código abierto especializada en la creación de interfaces gráficas web, permitiendo crear de forma rápida interfaces intuitivas para el usuario. Por otra parte, en cuanto diseño visual de la página, se ha optado utilizar MaterialUI[42] que utiliza el sistema de diseño de *Material Design* [43] de Google. Esta elección garantiza una experiencia de usuario consistente y profesional, ya que MaterialUI proporciona componentes estandarizados que siguen las mejores prácticas de diseño.

Por otro lado, se tiene la API que tiene tantos *endpoints* como número de enfoques implementados. Y la herramienta que se utiliza para el desarrollo de la API es FastAPI [44] que es una potente librería de Python que permite crear aplicaciones web de manera eficiente y robusta. Del mismo modo se implementa el buscador semántico que devolverá una consulta cuyos resultados contiene las características del refrán consultado, estos datos son almacenados en una base de datos vectorial siendo Pinecone[45] el utilizado para este trabajo.

Un aspecto técnico destacable en la arquitectura de la API es la implementación de Docker[46], una herramienta fundamental que proporciona aislamiento entre el entorno de desarrollo local y la API. Esta herramienta resuelve potenciales conflictos de versiones entre librerías y simplifica el proceso de despliegue en diferentes máquinas o servidores, asegurando la consistencia en las dependencias y la ejecución del sistema.

[41] https://github.com/facebook/react
[42] https://github.com/mui/material-ui
[43] https://m2.material.io/design/introduction
[44] https://github.com/fastapi/fastapi
[45] https://www.pinecone.io/
[46] https://www.docker.com/

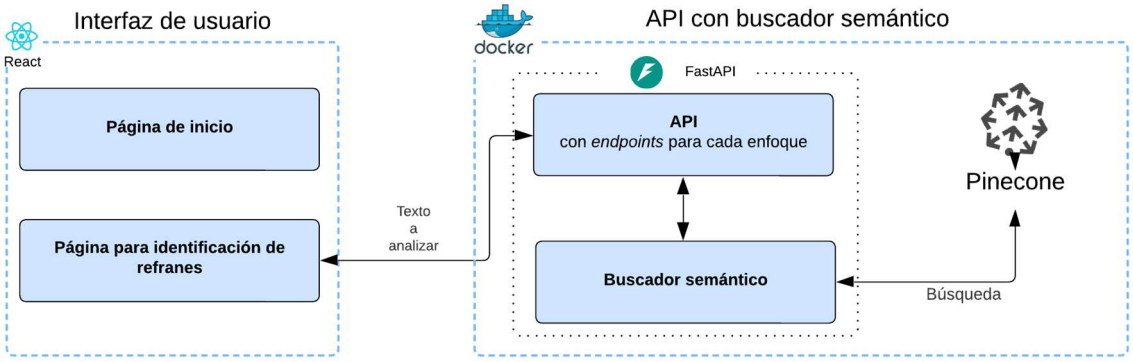

Figura 4.24 Arquitectura de la página web.

4.11.2 Desarrollo del Buscador Semántico

El desarrollo del buscador semántico se implementó con el objetivo específico de recuperar las características asociadas a los refranes cuando el sistema identifica un texto como refrán. Este componente es fundamental ya que, cuando un usuario ingresa un texto que el sistema reconoce como refrán, el buscador recupera y muestra automáticamente sus características detalladas en la interfaz de usuario.

Para lograr esta funcionalidad, se utiliza la biblioteca Sentence Transformers[47] con el modelo pre-entrenado 'all-MiniLM-L6-v2', que convierte los refranes almacenados y el texto de consulta en representaciones vectoriales, capturando así su significado semántico. Tanto los refranes como sus características asociadas (variantes, contextos, sinónimos, antónimos e ideas clave) se procesaron y almacenaron en Pinecone, una base de datos vectorial que facilita búsquedas eficientes por similitud semántica.

4.11.2.1 Pasos para Implementar el Buscador Semántico

A continuación, se describen detalladamente los pasos seguidos para implementar cada fase del buscador semántico, desde la preparación y carga inicial de datos hasta el desarrollo del servicio de consulta, incluyendo los aspectos técnicos y consideraciones relevantes en cada etapa del proceso.

En la Figura 4.25 se muestra un resumen de los dos pasos principales utilizados para la implementación mismos que se ramifican en sub pasos.

[47] https://github.com/UKPLab/sentence-transformers

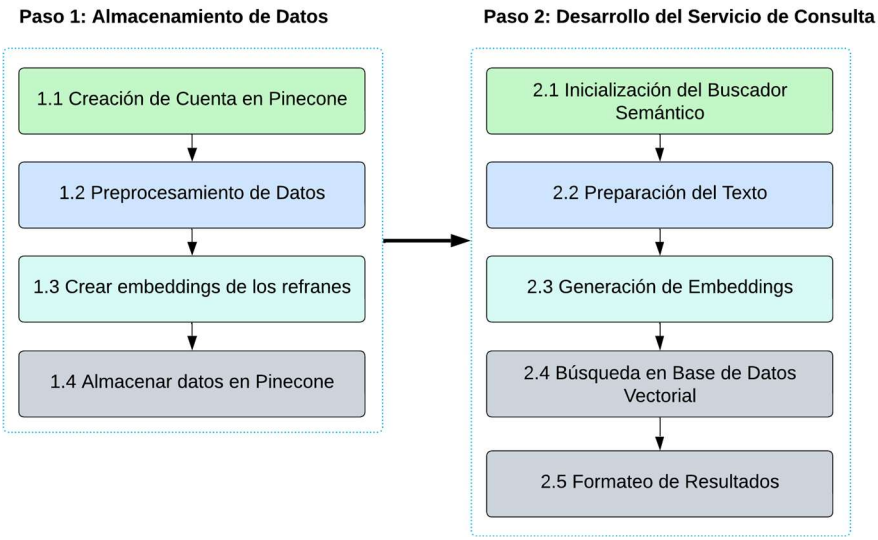

Figura 4.25 Pasos para implementar el buscador semántico.

Paso 1: Almacenar Datos a Pinecone

Como primer paso es necesario almacenar los datos de los refranes extraídos con sus respectivas características. Para esto se ha seguido los siguientes pasos:

Paso 1.1: Creación de Cuenta en Pinecone

Como paso inicial, se requiere crear una cuenta en la página oficial de Pinecone para obtener una llave secreta (*secret key*) que permite la autenticación necesaria para cargar y consultar datos en la base de datos vectorial. La elección de Pinecone se fundamenta en su plan gratuito, el cual ofrece características adecuadas para este proyecto: capacidad de almacenamiento de hasta 2GB, un límite de un millón de consultas mensuales y la posibilidad de mantener un proyecto activo. Estas especificaciones se ajustan perfectamente a los requerimientos del sistema, haciendo de Pinecone una solución óptima para la implementación del buscador semántico.

Paso 1.2: Preprocesamiento de Datos

Como siguiente paso es necesario limpiar los datos de los refranes, así como remover caracteres especiales, para este paso no debemos retirar signos de puntuación que son propios de los refranes ya que se busca tener una base de datos limpia que sirva para consultas.

Paso 1.3: Crear embeddings de los refranes

La siguiente fase consiste en la generación de *embeddings* para los enunciados de los refranes, un paso es importante para el funcionamiento del buscador semántico. Este proceso se centra específicamente en los enunciados, ya que la búsqueda se basa en encontrar similitudes semánticas entre el texto ingresado por el usuario en la interfaz y los enunciados almacenados. Para esta transformación, se utiliza el modelo pre-entrenado all-MiniLM-L6-v2, que

convierte cada enunciado en su correspondiente representación vectorial, permitiendo así comparaciones semánticas efectivas.

```
model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')
embeddings = model.encode(df["refran"], batch_size=64,
show_progress_bar=True)
df['embeddings'] = embeddings.tolist()
```

Paso 1.4: Almacenar datos en Pinecone

Como último paso para el almacenamiento de datos se procede a subir los datos a pinecode, teniendo tres campos principales que son el id, que identifica como registro único a cada refrán que se va a almacenar, el valor (*value*) que corresponde al embedding de los refranes siendo este por el cual se va hacer la búsqueda semántica y por último el campo los metadatos (*metadata)* que es donde se almacenan las características del refrán como es el significado, variantes, sinónimos y antónimos.

```
    to_upsert = data[['id','values','metadata']].to_dict('records')
    _ = index.upsert(to_upsert)
```

Para realizar consultas en el sistema, se implementa la interacción con la base de datos vectorial mediante el cliente de Pinecone. Esta implementación permite encontrar refranes similares al texto consultado mediante un sistema de puntuación de similitud, donde el proceso de búsqueda retorna un valor (score) que oscila entre 0 y 1. Este score es un indicador de la similitud entre el texto consultado y los refranes almacenados en la base de datos, donde un valor cercano a 0 indica una baja similitud, mientras que un valor cercano a 1 representa una coincidencia exacta o una alta similitud con algún refrán en la base de datos. En la Figura 4.26 se ilustra un ejemplo de respuesta.

```
'matches': [{'id': '816',
             'metadata': {'antonimos': [],
                          'contextos': [],
                          'ideas_claves': ['familia', 'ira'],
                          'marcador_de_uso': 'en desuso',
                          'observaciones': '',
                          'observaciones_lexicas': 'alheña es el polvo de las '
                                                   'hojas secas y molidas de '
                                                   'la hierba que lleva el '
                                                   'mismo nombre y que sirve '
                                                   'para teñir. alheñar '
                                                   'significa «teñir con '
                                                   'polvos de alheña».',
                          'refran': 'Ira de hermanos, ira de diablos',
                          'significado': 'advierte de la gravedad que pueden '
                                         'alcanzar los efectos de la '
                                         'enemistad y el rencor entre '
                                         'hermanos, y también entre personsa '
                                         'unidas por parentesco.',
                          'sinonimos': ['pelea de hermanos, alheña en manos'],
                          'variantes': []},
             'score': 0.572727144,
             'values': []}],
 'namespace': '',
 'usage': {'read_units': 6}}
```

Figura 4.26 Ejemplo de resultado de consulta de Pinecone.

Paso 2: Consulta de la Base de Datos Vectorial

Este paso es utilizado e implementado como servicio en la API de este trabajo, para esto se sigue los siguientes pasos:

Paso 2.1: Inicialización del Buscador Semántico

Como primer paso se inicializa con los componentes necesarios: el modelo de *embeddings* pre-entrenado, el nombre del índice de Pinecone, la clave API y un limpiador de texto. Durante la inicialización, se establece la conexión con Pinecone utilizando la clave API proporcionada.

```
model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')
self.pc = pinecone.Pinecone(api_key=api_key)
self.index_pc = self.pc.Index(self.index_name)
```

Paso 2.2: Preparación del Texto

Cuando se recibe una consulta, el primer paso es limpiar el texto. Este proceso elimina elementos no deseados y estandariza el formato del texto para mejorar la precisión de la búsqueda.

Paso 2.3: Generación de Embeddings

El texto limpio se convierte en un vector de *embeddings* utilizando el modelo pre-entrenado. Este proceso transforma el texto en una representación numérica que captura su significado semántico.

Paso 2.4: Búsqueda en la Base de Datos Vectorial

Se realiza una consulta a Pinecone utilizando el vector generado. La búsqueda retorna los refranes más similares semánticamente, limitando los resultados según el parámetro *top_k* especificado.

```
response = self.index_pc.query(
            vector=query_vector,
            top_k=top_k,
            include_metadata=True,
            include_values=False,
        )
```

Para propósitos de este trabajo el parámetro *top_k* es de 1 ya que únicamente devolveremos la primera coincidencia y también filtraremos los que tengan una similitud del 80%.

Paso 2.5: Formateo de Resultados

Como último paso se procesan y formatean en una estructura de datos más amigable, incluyendo todas las características relevantes del refrán: enunciado, sinónimos, antónimos, variantes, significado, contextos, ideas clave y otros metadatos importantes. En la Figura 4.27 se muestra un ejemplo de la respuesta.

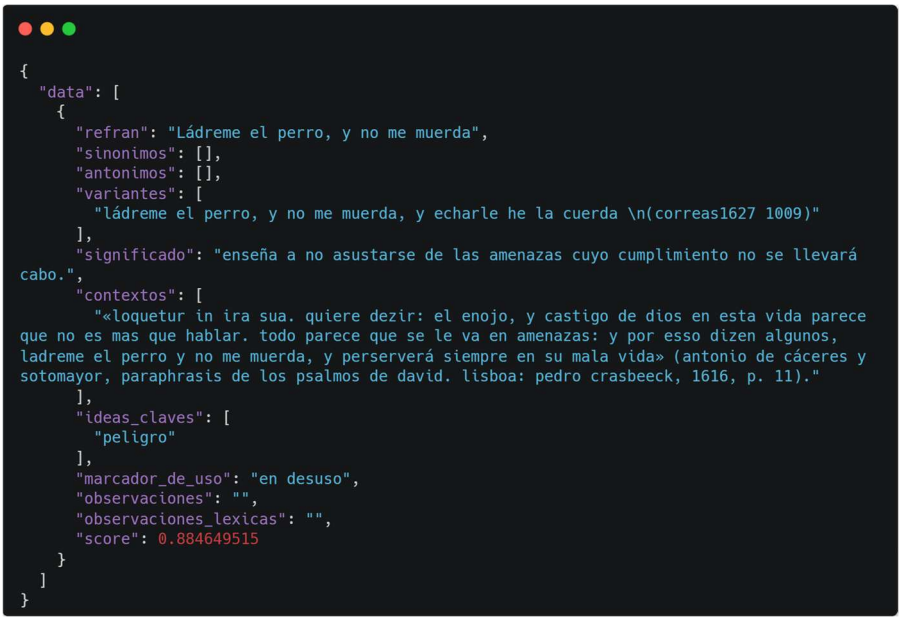

Figura 4.27 Ejemplo de respuesta del buscador semántico.

4.11.3 Desarrollo de la API

En esta sección se detalla el proceso de desarrollo de la API que permite acceder a los diferentes enfoques de identificación de refranes mediante *endpoint*s específicos para cada enfoque.

Como se ilustra en la Figura 4.28, cada *endpoint* de la API esta identificada por el nombre del enfoque en su *url*, el cual al recibir una solicitud HTTP (*Hypertext Transfer Protocol* por sus siglas en inglés) del tipo *POST,* realiza los siguientes pasos:

1. La API recibe el texto de entrada que va a analizar.
2. Se identifica y selecciona el enfoque que está determinado por la dirección del *endpoint*.
3. Se realiza el análisis del refrán utilizando el enfoque seleccionado. Este proceso incluye el preprocesamiento del texto y la aplicación de los procedimientos específicos del enfoque elegido (descritos en secciones anteriores de este capítulo).
4. Si el sistema identifica el texto como ordinario (no refrán), devuelve una respuesta al cliente con una etiqueta negativa (False). En caso de identificarlo como refrán, continúa con el proceso.
5. Para los textos identificados como refranes, el sistema realiza una búsqueda en la base de datos mediante el buscador semántico.
6. Al verificar la existencia del refrán en la base de datos (DB), si no se encuentra coincidencia, se devuelve una etiqueta positiva (True) al cliente, sin incluir información adicional sobre su significado.
7. Si existe el refrán en la base de datos se procede a adaptar el significado de dicho refrán a lectura fácil.
8. Como paso final, cuando el texto ha sido identificado como refrán y encontrado en la base de datos, la respuesta al cliente incluye: el significado

adaptado a lectura fácil, las posibles variantes del refrán si esta las tuviere y una etiqueta positiva (True) que identifica al texto como refrán.

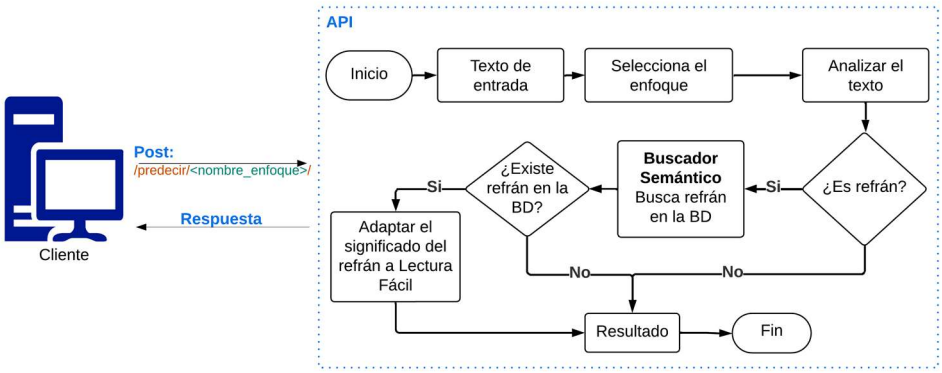

Figura 4.28 API – Procesamiento de texto para la identificación de refranes.

En lo que respecta a la respuesta de la API esta tiene un formato JSON como se muestra en Figura 4.29, para todos los enfoques se tiene la misma estructura:

- Nombre del modelo (*model_name*): identifica el nombre del modelo que se utiliza para la identificación del refrán.
- Texto (*text*): el texto preprocesado que va a ser analizado.
- Etiqueta (*label)*: identifica si el texto analizado es refrán (*True)* o es un texto ordinario (*False*).
- Significado (*meaning*): el significado del refrán si este estuviera presente en la base de datos.
- Variantes (*variants):* variantes del refrán.


```
{
  "model_name": "gpt-4o-mini",
  "saying": "dios ayuda al que madruga",
  "label": true,
  "meaning": "Recomienda ser diligente para para tener éxito en las pretensiones, en el
trabajo.",
  "variants": [
    {
      "id": 1,
      "proverb": "A quien madruga, dios lo ayuda"
    },
    {
      "id": 2,
      "proverb": "Más hace el que dios ayuda que el que mucho madruga"
    },
    {
      "id": 3,
      "proverb": "Más vale a quien dios ayuda que quien mucho madruga"
    },
    {
      "id": 4,
      "proverb": "Al que madruga dios lo ayuda"
    }
  ]
}
```


Figura 4.29 Ejemplo de respuesta en formato JSON de la API para la identificación de refranes.

Como se mencionó anterior mente la API permite acceder a los diferentes enfoques como se detalla en la siguiente Tabla 4.17.

Endpoint	Modelo	Descripción	Corpus
/saying/t5_base_01/	FLAN-T5-base	Primer experimento sin utilizar prefijos	Corpus 2
/saying/t5_small_02/	FLAN-T5-small	Primer experimento sin utilizar prefijos	Corpus 2
/saying/t5_large_03/	FLAN-T5-large	Primer experimento sin utilizar prefijos	Corpus 2
/saying/t5_small_04/	FLAN-T5-small	Segundo experimento utilizando prefijo "clasificar"	Corpus 2
/saying/t5_base_05/	FLAN-T5-base	Segundo experimento utilizando prefijo "clasificar"	Corpus 2
/saying/t5_large_06/	FLAN-T5-large	Segundo experimento utilizando prefijo "clasificar" y formato de salida específico	Corpus 2
/saying/t5_small_07/	FLAN-T5-small	Tercer experimento utilizando prefijo "clasificar"	Corpus 3
/saying/t5_small_08/	FLAN-T5-small	Cuarto experimento utilizando prefijo "clasificar"	Corpus 4
/saying/t5_small_09/	FLAN-T5-small	Quinto experimento utilizando prefijo "clasificar"	Corpus 5
/saying/t5_small_10/	FLAN-T5-small	Sexto experimento utilizando prefijo "clasificar" y características de rimas	Corpus 5
/saying/gpt/	OpenAI GPT	Uso de librería de OpenAI para interacción con la API	Corpus 2
/saying/rl_4_2_1/	Regresión Logística	Red convolucional con características [CM1]	Corpus 1
/saying/rl_4_2_2/	Regresión Logística	Red convolucional con características [CM8, CM12] y [CM18-CM20]	Corpus 5
/saying/rf_4_3_1/	Random Forest	Red convolucional con características [CM8, CM12] y [CM18-CM20]	Corpus 5
/saying/svm_4_4_1/	SVM	Red convolucional con características [CM8, CM12] y [CM18-CM20]	Corpus 5

Tabla 4.17 *Endpoints* de la API con su correspondiente descripción por enfoque.

4.11.4 Desarrollo de la Interfaz Gráfica de Usuario

Para concluir con este capítulo hablaremos del desarrollo de la interfaz gráfica de usuario. Este componente de la página web permite al usuario interactuar de forma intuitiva con los diferentes enfoques desarrollados.

Para esto se ha utilizado React y Material UI para el desarrollo de los componentes de la interfaz.

4.11.5 Requisitos de Accesibilidad

Para el desarrollo de la interfaz gráfica se ha utilizado los lineamientos de lectura fácil [78] el cual definen una serie de requerimientos y que se los adoptará para el diseño de la interfaz gráfica. Entre los requerimientos que se ha implementado en la interfaz son los siguientes:

- **Claridad en la navegación:** El contenido se presentarán de manera clara y con lenguaje accesible, esto considerando las pautas de la iniciativa sobre Accesibilidad de la Web (*Web Accessibility Initiative, WAI*)[48]. Entre la principal que se ha implementado en nuestra página web es el uso de etiquetas para controles de formulario, entrada y otros componentes de la interfaz de usuario.

- **Evitar distracciones:** Evitar efectos tipográficos, como adornos, colores y sombras.

- **Características tipográficas:**
  - Alinear el texto a la izquierda, no justificándolo a la derecha.
  - Utilizar dos tipos de textos como máximo: para textos y para títulos.
  - El tamaño de letra debe ser suficientemente grande, entre 12 y 16 puntos.
  - Utilizar negritas y subrayados para destacar palabras, aunque siempre de forma moderada para evitar distracciones.
  - Utilizar un interlineado acorde a la tipografía, que puede variar desde sencillo a más amplio (1,3 a 1,5 veces mayor que el espacio medio entre palabras o 30% del tamaño de la letra).
  - Cada línea debe tener una sola oración, preferentemente, aunque también puede contener dos como máximo.
  - Utilizar tipografías sin remate, por ser más claras. Entre otras, se pueden destacar Arial, Calibri, Candara, Corbel, Gill Sans, Helvetica, Myriad, Segoe, Tahoma, Tiresias y Verdana.

[48] https://www.w3.org/WAI/

4.11.6 Vistas de la Página Web

La página web se compone de dos vistas: la página de inicio y página de identificación de refranes.

4.11.6.1 Página de Inicio

La página de inicio (ver Figura 4.30) se compone de varias partes: Una cabecera con el nombre de la aplicación y una sección que muestra el título del proyecto y su año de finalización y el nombre del autor. Además, esta página tiene un botón “Iniciar” que redireccionará a la página de identificación de refranes.

Figura 4.30 Página de inicio.

4.11.6.2 Página de Identificación de Refranes

Como se muestra en la Figura 4.31 esta página ofrece al usuario la identificación de refranes en texto en español, teniendo como principales componentes:

1. Botón para regresar a la página de inicio.
2. Menú desplegable para seleccionar los diferentes enfoques.
3. Un campo de texto donde el usuario ingresará el texto a identificar.
4. Un botón para limpiar el campo de texto.
5. Un botón que inicia el análisis del texto, para identificar si el texto ingresado es un refrán o un texto ordinario.

Figura 4.31 Página de identificación de refranes: Componentes de la interfaz.

El usuario puede ingresar el texto a identificar en el campo de texto, seguido de eso seleccionar el enfoque utilizando el menú desplegable y finalizaría iniciando la identificación del texto utilizando el botón "Analizar texto". Como respuesta el sistema determinará si el texto ingresado es un refrán o no.

Si el texto ingresado es un refrán se mostrarán la etiqueta correspondiente y la explicación de dicho refrán con sus respectivas variantes que estén registradas en la base de datos, como se muestra en la Figura 4.32.

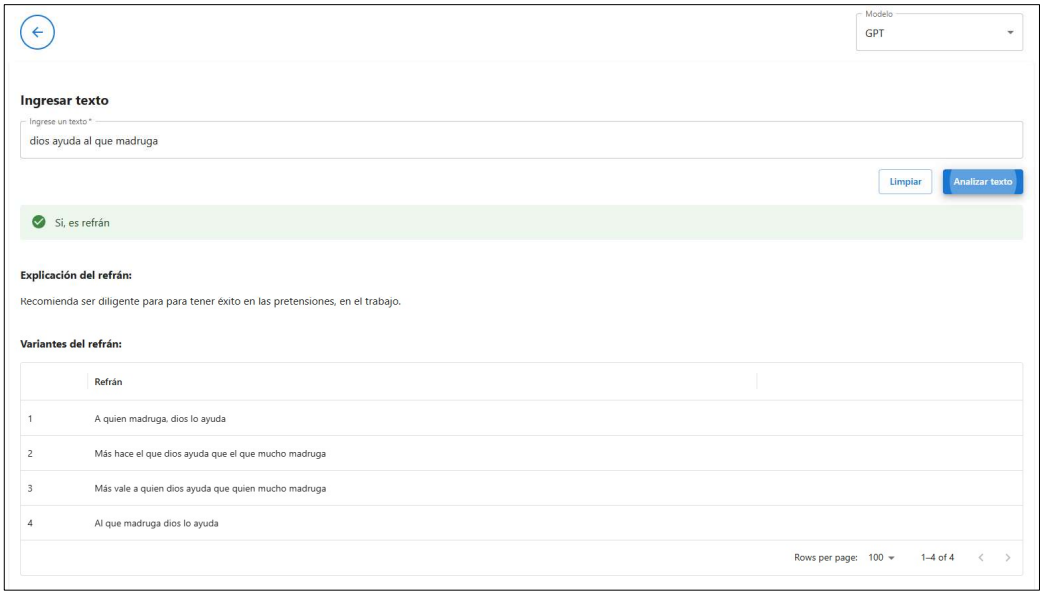

Figura 4.32 Página de identificación de refranes: Texto identificado como refrán.

En el caso contrario de no identificar el texto como un refrán válido mostrará un mensaje "No, es refrán", como se muestra en la Figura 4.33.

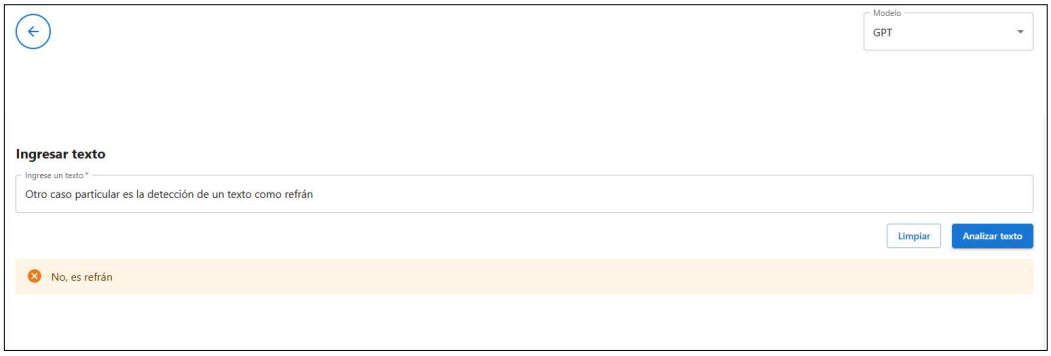

Figura 4.33 Página de identificación de refranes: Texto identificado como no refrán.

Otro caso particular es la detección de un texto como refrán, pero que dicho refrán no se encuentre en nuestra base de datos, en este caso no se mostrarán las variantes del posible refrán identificado, como se muestra en la Figura 4.34.

Figura 4.34 Página de identificación de refranes: Texto identificado como refrán y no registrado en la base de datos.

Otra característica de esta vista es el manejo de errores cuando el texto ingresado es un texto vacío Figura 4.35, y el aviso de la conexión con el servidor de la API como se muestra en la Figura 4.36.

Figura 4.35 Página de identificación de refranes: Manejo de errores al ingresar un texto vacío.

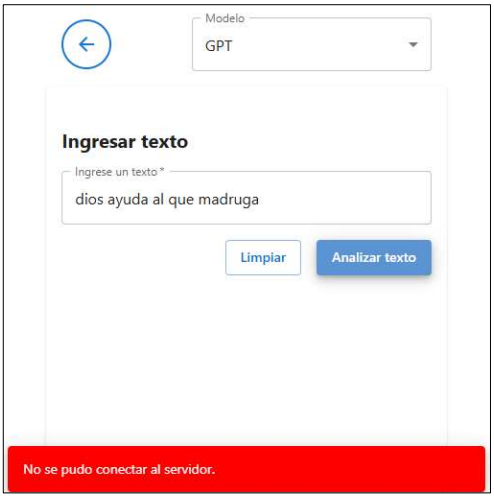

Figura 4.36 Página de identificación de refranes: Manejo de errores al no tener respuesta de la API.

5 Resultados y Discusión

En esta sección se presentan los resultados obtenidos en la identificación de refranes utilizando distintos enfoques, incluyendo regresión logística, modelos basados en redes convolucionales con el uso de FastText para la vectorización, así como modelos de lenguaje preentrenados de la familia FLAN-T5 y GPT-4o-mini.

Para evaluar el rendimiento de los enfoques propuestos, se utilizaron las siguientes métricas de evaluación.

- **Precisión (*Precision*):** Proporción de identificaciones correctas entre todas las predicciones positivas.
- **Exhaustividad (*Recall*):** Proporción de refranes correctamente identificados del total de refranes reales.
- **Puntaje F1 (*F1-Score*):** Media entre precisión y exhaustividad.
- **Exactitud (*Accuracy*):** Medida de cuán cerca del valor real se encuentra el valor medido.
- **Matriz de confusión:** Visualización de predicciones correctas e incorrectas.
    - **Verdadero Positivo (VP):** Representa los casos en el que se ha identificado correctamente un refrán.
    - **Falso Positivo (FP):** Indica los casos en el que se ha identificado erróneamente una oración normal como refrán.
    - **Falso Negativo (FN):** Representa los casos en el que se identifica un refrán como una oración normal.
    - **Verdadero Negativo (VN):** Indica los casos en el que se identifica correctamente una oración normal.

En lo que corresponden a los datos de entrenamiento, para este estudio se han utilizados los cinco corpus que se construyeron en secciones anteriores, que están compuestos por oraciones y refranes en español como se muestra en la Tabla 5.1. Cada corpus varía en tamaño y diversidad lingüística, con el objetivo de evaluar el impacto de las distintas variaciones de corpus en el rendimiento de los modelos.

Corpus	Oraciones (Original)	Refranes (Original)	Total (Original)
Corpus 1	6,713	3,958	10,671
Corpus 2	6,674	4,050	10,724
Corpus 3	12,723	9,833	22,556
Corpus 4	14,370	9,851	24,221
Corpus 5	20,577	9,824	30,401

Tabla 5.1 Composición de los corpus antes del balanceo.

Se observa un desbalance entre oraciones y refranes, lo que ocasionaría un sesgo en el entrenamiento de los modelos, favoreciendo la identificación a la clase mayoritaria (oraciones) y afectando a la detección de refranes.

Para mejorar la capacidad de los modelos en la identificación de refranes, se aplicó una estrategia de submuestreo y sobre muestreo, permitiendo obtener conjuntos balanceados, con igual cantidad de oraciones y refranes, como se muestra en la Tabla 5.2.

Corpus	**Oraciones (Balanceado)**	**Refranes (Balanceado)**	**Total (Balanceado)**
Corpus 1	5,000	5,000	10,000
Corpus 2	5,000	5,000	10,000
Corpus 3	9,000	9,000	18,000
Corpus 4	10,000	10,000	20,000
Corpus 5	15,000	15,000	30,000

Tabla 5.2 Composición de los corpus después del balanceo.

Cada corpus fue dividido en tres subconjuntos para ser usados en los diferentes enfoques, como se muestra en la Tabla 5.3.

Corpus	**Entrenamiento**	**Validación**	**Prueba**
Corpus 1	10,000	1,192	1,325
Corpus 2	10,000	1,192	1,318
Corpus 3	18,000	2,507	2,785
Corpus 4	20,000	2,692	2,991
Corpus 5	30,000	3,378	3,754

Tabla 5.3 Distribución de los corpus en entrenamiento, validación y prueba

Para evaluar el rendimiento del modelo en un entorno de producción, se utilizó un conjunto de datos compuesto por oraciones diversas, algunas de estas corresponden a refranes y otras a textos generales que tienen una estructura similar a un refrán (ver Tabla 5.4), su uso en la evaluación en producción proporciona una visión realista del desempeño del modelo en escenarios no controlados.

Oración	Etiqueta
Bajo la casa, está la terraza	0
Juan de Dios salió a trabajar en la madrugada	0
Juan de dios salió de paseo en la madrugada con Mateo	0
Pepe Pecas madruga para que le vaya bien en el trabajo	0
Más sabe el ladrón por viejo que por ladrón	1
A la piedra piedra, y al zapato la piedra	0
Del teclado a la pantalla, el error nunca calla	1
Más vale código probado que desarrollador confiado	1
Backup temprano, problema lejano	1
En casa del programador, cada variable es un valor	1
Del dicho al código, hay un largo trecho	1
En casa de informático, ordenador lleno de virus	1
Ingeniero sin panza no es de confianza	1
Por la mañana, café, por la tarde, también café	0
No hay mejor medicina que dormir hasta mediodía	0
Al pan con mantequilla, más mantequilla todavía	0
Escoba nueva barre bien	1
Su temperamento es particularmente inestable: a veces ríe, a veces llora.	0
Jorge madrugó en la mañana, pero su amigo Dios no lo esperó	0
El diablo le preguntó al viejo si sabía dónde estaba la dirección	0
Él tomó vino y comió pan.	0
El alma repartida, el alma rota	0
De luces encendidas y en la nieve	0
Reviven, con olor, y luego vuelven	0
A ser en la memoria igual que un eco	0
Crecen las flores. Dormiré un momento	0
Verde, amarilla, gris, blanca en la cumbre	0
Eternamente enaltecida y mansa	0
A veces voy por un camino	0
Y el camino se alarga, como el miedo a estar vivo	0
He vuelto a creer en Dios	0
Con sus piedras y sus matas secas	0
A lluvia de muy lejos. Suena esa lluvia. Y pienso sin ganas	0
Pocas palabras necesarias, y quitarse de en medio	0
Y los trigos en éxtasis de Castilla la Vieja	0

Tabla 5.4 Corpus para validar los enfoques en producción.

5.1 Resultados de Enfoques Basado en Clasificador de Aprendizaje Automático

Para la obtención de los resultados en la tarea de identificación de refranes se emplearon el Corpus 1 y Corpus 2, siendo el primer corpus compuesto por los refranes del Centro Virtual Cervantes, mientras que el Corpus 2 tiene la misma composición, pero se diferencia del primero al no haber lematizado sus oraciones, ya que dicho proceso puede eliminar las rimas de los refranes, afectando la identificación de refranes.

Para la experimentación, se utilizó en todos los casos regresión logística con una representación del texto basada en vectorización de frecuencia de términos (*Term Frequency Vectorization*). En la Tabla 5.5 se presentan los experimentos realizados, diferenciados por el corpus utilizado y las características consideradas.

	Corpus	**Características**
Experimento 1	Corpus 1	CM1 - CM12
Experimento 2	Corpus 1	CM1 - CM12, CM13 - CM17
Experimento 3	Corpus 2	CM1 - CM12
Experimento 4	Corpus 2	CM1 - CM12, CM13 - CM17

Tabla 5.5 Lista de experimentos realizados con regresión logística.

Las características utilizadas se dividen en dos grupos:

- CM1 a CM12: Medidas basadas en conteo de palabras y frecuencias.
- CM13 a CM17: Características relacionadas con emociones en el texto.

Con estos experimentos se detectaron problemas de sobreajuste (*overfitting*) respecto a la longitud de las oraciones. A pesar de obtener un *accuracy* alto de entre 85 y 91% en las pruebas iniciales. Al evaluar los modelos en entornos reales con datos de usuarios, se observó que los modelos tienden a clasificar oraciones cortas como refranes, mientras que las oraciones más largas son clasificadas incorrectamente como no refranes.

Para validar este comportamiento se utiliza un pequeño corpus compuesto por 9 refranes y 26 oraciones que no son refranes (35 oraciones en total) con el objetivo de verificar este comportamiento de los modelos en este escenario.

5.1.1 Resultados de los Experimentos

A continuación, se describe los resultados obtenidos para cada experimento.

Experimento 1: Corpus 1 con características CM1 – CM12

Para este experimento se obtuvo la matriz de confusión mostrada en la Figura 5.1, el cual muestra que el modelo identifica erróneamente la mayoría de los textos como no refrán (falsos negativos).

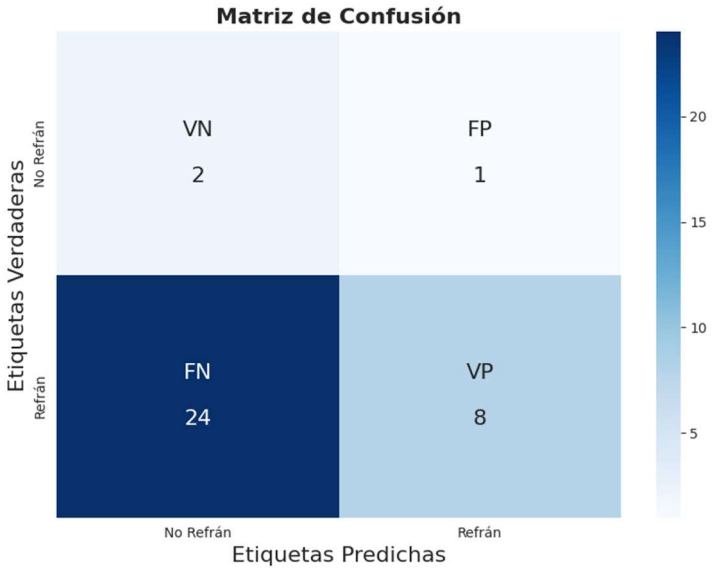

Figura 5.1 Experimento 1: Corpus 1 con características CM1 – CM12 (matriz de confusión).

En este experimento se obtuvo un *accuracy* del 29% como se muestra en la Tabla 5.6, lo que evidencia una baja capacidad de generalización del modelo.

Los resultados también muestran que el modelo tiene una exhaustividad (*recall*) bajo en la detección de refranes siendo este del 25%, lo que indica que la mayoría de los refranes no fueron identificados correctamente.

	Precision	Recall	F1-score	Support
No Refrán	0.08	0.67	0.14	3
Refrán	0.89	0.25	0.39	32
Accuracy			0.29	35
Macro avg	0.48	0.46	0.26	35
Weighted avg	0.82	0.29	0.37	35

Tabla 5.6 Experimento 1: Corpus 1 con características CM1 – CM12 (informe de clasificación).

Experimento 2: Corpus 1 con Características CM1 - CM12 y CM13 – CM17

En este caso se incorpora características emocionales, y como se muestra en la matriz de confusión de la Figura 5.2, se puede observar que se mantiene el modelo clasificando erróneamente la mayor parte de los textos como no refranes.

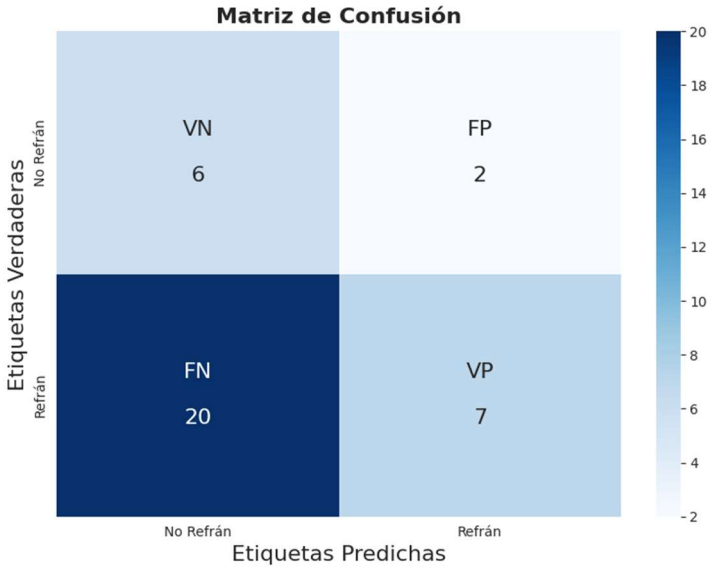

Figura 5.2 Experimento 2: Corpus 1 con Características CM1 - CM12 y CM13 - CM17 (matriz de confusión).

En la Tabla 5.7 se muestra un *accuracy* de 37%, mostrando una ligera mejora en con respecto al experimento 1. En este experimento se mantiene con una exhaustividad (*recall*) baja en la detección de refranes siendo este del 26%.

	Precision	Recall	F1-score	Support
No Refrán	0.23	0.75	0.35	8
Refrán	0.78	0.26	0.39	27
Accuracy			0.37	35
Macro avg	0.50	0.50	0.37	35
Weighted avg	0.65	0.37	0.38	35

Tabla 5.7 Experimento 2: Corpus 1 con Características CM1 - CM12 y CM13 - CM17 (informe de clasificación).

Experimentos 3: Corpus 2 (Sin Lematización) con características CM1 – CM12

En este experimento, se utilizó Corpus 2, donde los textos no fueron lematizados. Teniendo como resultado de las pruebas la matriz de confusión de la Figura 5.3.

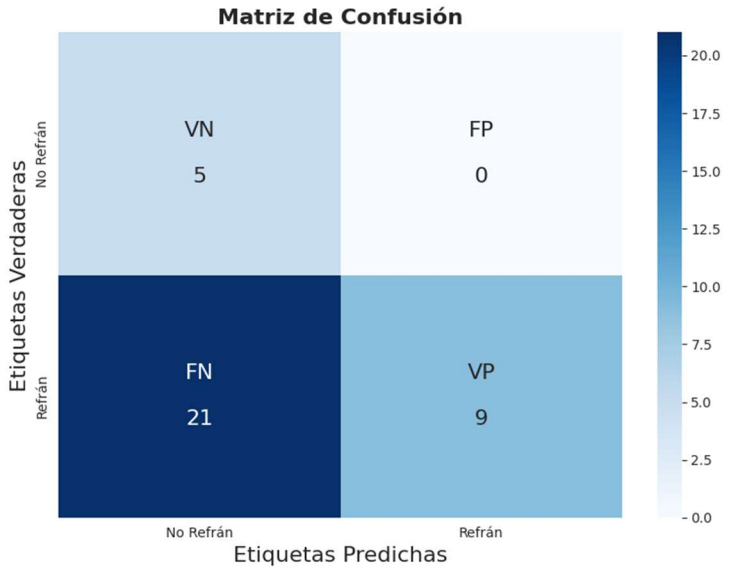

Figura 5.3 Experimentos 3: Corpus 2 (Sin Lematización) con características CM1 – CM12 (matriz de confusión).

En Tabla 5.8 la se muestra que se obtuvo un *accuracy* del 40%, lo que sugiere una ligera mejora al preservar las rimas.

La exhaustividad (*recall*) en refranes aumentó a 30%, lo que indica que eliminar la lematización puede ayudar a preservar información relevante para la identificación de refranes.

	Precision	Recall	F1-score	Support
No Refrán	0.19	1.00	0.32	5
Refrán	1.00	0.30	0.46	30
Accuracy			0.40	35
Macro avg	0.60	0.65	0.39	35
Weighted avg	0.88	0.40	0.44	35

Tabla 5.8 Experimentos 3: Corpus 2 (Sin Lematización) con características CM1 – CM12 (informe de clasificación).

Experimento 4: Corpus 2 (Sin Lematización) con Características CM1 - CM12 y CM13 - CM17

Este experimento utilizó características emocionales (CM13 - CM17) en el Corpus 2, sin aplicar lematización, y teniendo como resultado la matriz de confusión de la Figura 5.4.

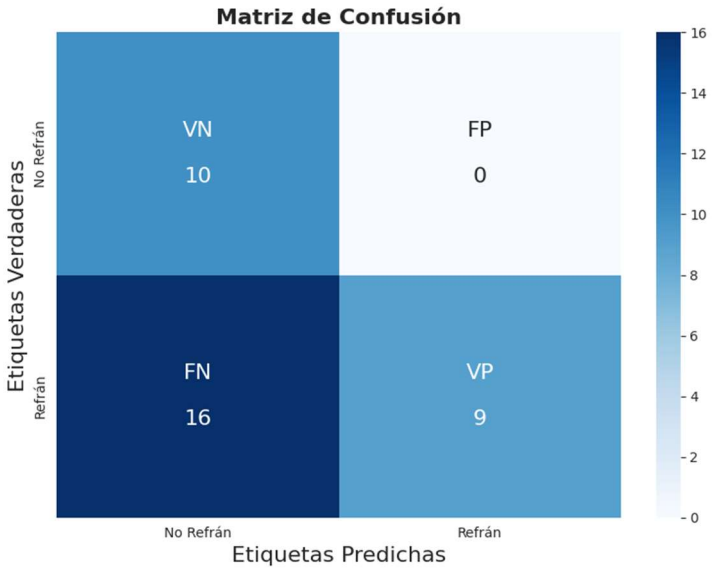

Figura 5.4 Experimento 4: Corpus 2 (Sin Lematización) con Características CM1 - CM12 y CM13 - CM17 (matriz de confusión)

Para este experimento como se observa en la Tabla 5.9 se obtuvo el mejor rendimiento hasta el momento, con un *accuracy* de 54%.

Este resultado sugiere que la preservación de la rima y el uso de características emocionales contribuyen a mejorar la detección de refranes.

	Precision	Recall	F1-score	Support
No Refrán	0.38	1.00	0.56	10
Refrán	1.00	0.36	0.53	25
Accuracy			0.54	35
Macro avg	0.69	0.68	0.54	35
Weighted avg	0.82	0.54	0.54	35

Tabla 5.9 Experimento 4: Corpus 2 (Sin Lematización) con Características CM13 - CM17 (informe de clasificación)

5.2 Resultados de Enfoques Híbridos

Para mejorar la identificación de refranes, se utilizó la vectorización de textos con FastText en combinación con distintos modelos de aprendizaje automático, incluyendo regresión logística, *Random Forest* y *Support Vector Machine*.

5.2.1 Regresión Logística con CNN

En lo que corresponde a la regresión logística, se ha realizado dos experimentos utilizando el mismo Corpus 5. Esto es con el objetivo de verificar el impacto de las características seleccionadas.

En la Tabla 5.5 se presentan los experimentos realizados, diferenciados las características consideradas en cada caso.

	Corpus	**Características**
Experimento 1	Corpus 5	CM1 - CM12
Experimento 2	Corpus 5	CM8 – CM12, CM18 – CM20

Tabla 5.10 Lista de experimentos realizados con regresión logística.

Las características utilizadas se dividen en dos grupos principales:

- CM1 - CM12: Medidas basadas en el conteo de palabras y frecuencias.
- CM8 - CM12 y CM18 - CM20:
    - CM8 - CM12: Características relacionadas con la frecuencia de palabras y sinónimos.
    - CM18 - CM20: Características asociadas a la rima, lo que permite evaluar su impacto en la identificación de refranes.

Estos experimentos permitirán determinar la influencia de las diferentes características en el rendimiento del modelo y su capacidad para identificar refranes de manera efectiva.

5.2.1.1 Resultados de los Experimentos

A continuación, se describe los resultados obtenidos para cada experimento.

Experimento 1: Corpus 5 con características CM1 – CM12

En este primer experimento, se utilizaron las características CM1 - CM12, correspondientes a medidas basadas en el conteo de palabras y frecuencias. La Figura 5.5 muestra la matriz de confusión obtenida en la evaluación del modelo.

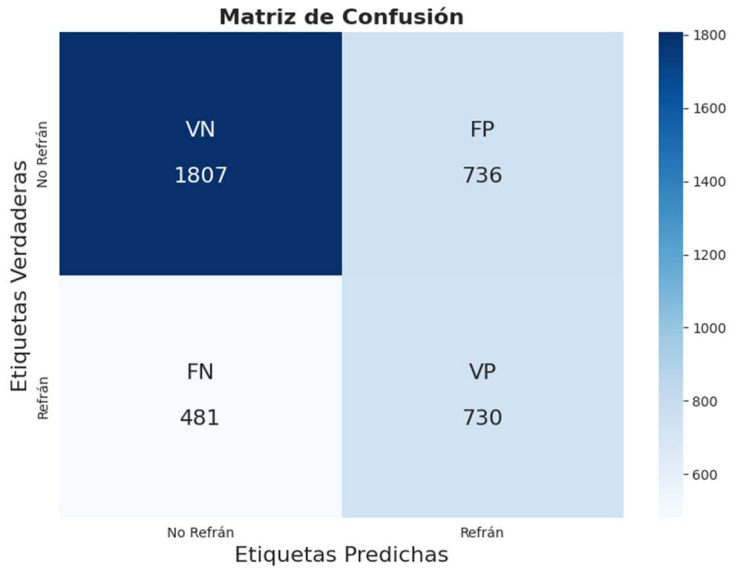

Figura 5.5 Experimento 1: Corpus 5 con características CM1 – CM12 (matriz de confusión).

Los resultados detallados del informe de clasificación se presentan en la Tabla 5.6.

	Precision	Recall	F1-score	Support
No Refrán	0.79	0.71	0.75	2543
Refrán	0.50	0.60	0.55	1211
Accuracy			0.68	3754
Macro avg	0.64	0.66	0.65	3754
Weighted avg	0.70	0.68	0.68	3754

Tabla 5.11 Experimento 1: Corpus 5 con características CM1 – CM12 (informe de clasificación).

Los resultados muestran que el modelo logró un *accuracy* del 68%, con una mejor precisión en la clasificación de oraciones que no son refranes (79%) en comparación con los refranes (50%). El *recall* para identificar refranes alcanzó un 60%, lo que indica que el modelo identificó correctamente el 60% de los refranes reales, pero aún tiene margen de mejora en esta tarea.

Experimento 2: Corpus 5 con características CM8 – CM12 y CM18 – CM20

En este experimento, se incorporaron las características CM18 - CM20, que incluyen información sobre rima, junto con las características de conteo de palabras y frecuencias CM8 - CM12. La Figura 5.6 muestra la matriz de confusión obtenida.

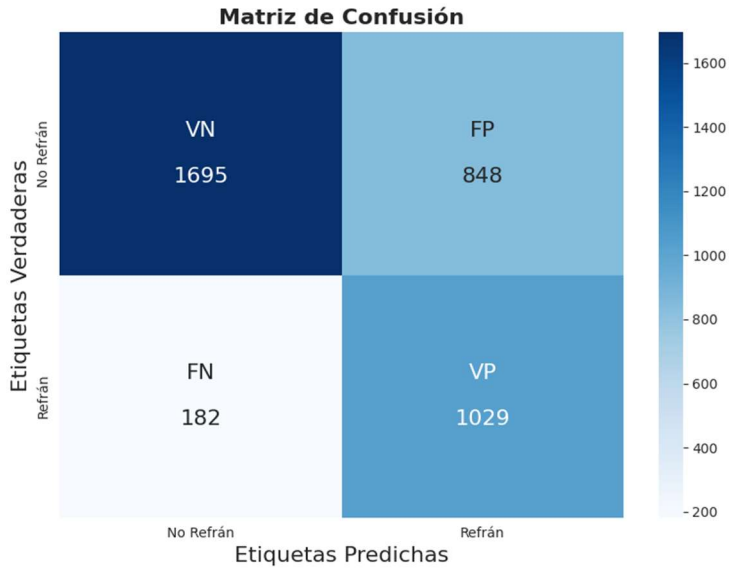

Figura 5.6 Experimento 2: Corpus 5 con características CM8 – CM12 y CM18 – CM20 (matriz de confusión).

Los resultados del informe de clasificación se presentan en la Tabla 5.12.

	Precision	Recall	F1-score	Support
No Refrán	0.90	0.67	0.77	2543
Refrán	0.55	0.85	0.67	1211
Accuracy			0.73	3754
Macro avg	0.73	0.76	0.72	3754
Weighted avg	0.79	0.73	0.73	3754

Tabla 5.12 Experimento 2: Corpus 5 con características CM8 – CM12 y CM18 – CM20 (informe de clasificación)

Este modelo obtuvo un *accuracy* del 73%, mostrando una mejora respecto al Experimento 1. Se observa un incremento significativo en la capacidad de detección de refranes, con un *recall* del 85%, lo que indica que el modelo identificó correctamente una mayor proporción de refranes. Además, la precisión en la clasificación de refranes aumentó a 55%, mientras que la precisión en la detección de oraciones que no son refranes se mantuvo alta (90%).

5.2.2 Random Forest con CNN

En lo que corresponde al uso de *Random Forest* con una red convolucional se realiza un experimento utilizando el Corpus 5 en conjunto con las características CM8 - CM12 y CM18 - CM20. La Figura 5.6 muestra la matriz de confusión obtenida.

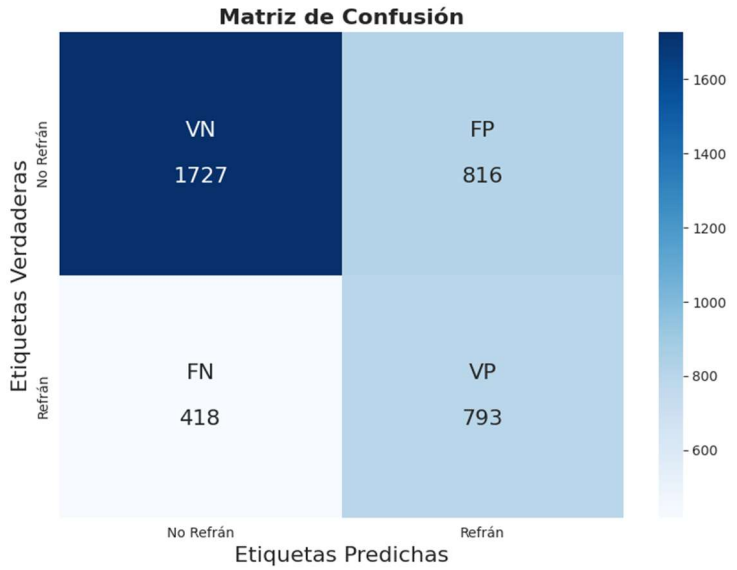

Figura 5.7 *Random Forest* con CNN (matriz de confusión).

Los resultados del informe de clasificación se presentan en la Tabla 5.13.

	Precision	Recall	F1-score	Support
No Refrán	0.81	0.68	0.74	2543
Refrán	0.49	0.65	0.56	1211
Accuracy			0.67	3754
Macro avg	0.65	0.67	0.65	3754
Weighted avg	0.70	0.67	0.68	3754

Tabla 5.13 *Random Forest* con CNN (informe de clasificación).

El modelo presenta un *accuracy* del 67%, lo que indica un desempeño moderado en la identificación de refranes. La precisión para las oraciones que no son refranes fue de 81%, mientras que la precisión para los refranes fue menor (49%), lo que sugiere que el modelo aún presenta dificultades en la identificación precisa de refranes.

Por otro lado, el *recall* para los refranes fue del 65%, lo que indica que el modelo logra recuperar una cantidad considerable de refranes reales, aunque con una precisión limitada. El F1-score promedio se mantiene en 0.65, lo que sugiere que el modelo tiene un equilibrio moderado entre precisión y *recall* en ambas clases (refranes y no refranes).

5.2.3 SVM con CNN

Como último enfoque híbrido, se evaluó el desempeño de la combinación de Máquinas de Soporte Vectorial (SVM) con una red convolucional (CNN) en la tarea de identificación de refranes. Para ello, se utilizó el Corpus 5, en conjunto

con las características CM8 - CM12 y CM18 - CM20. La Figura 5.8 muestra la matriz de confusión obtenida.

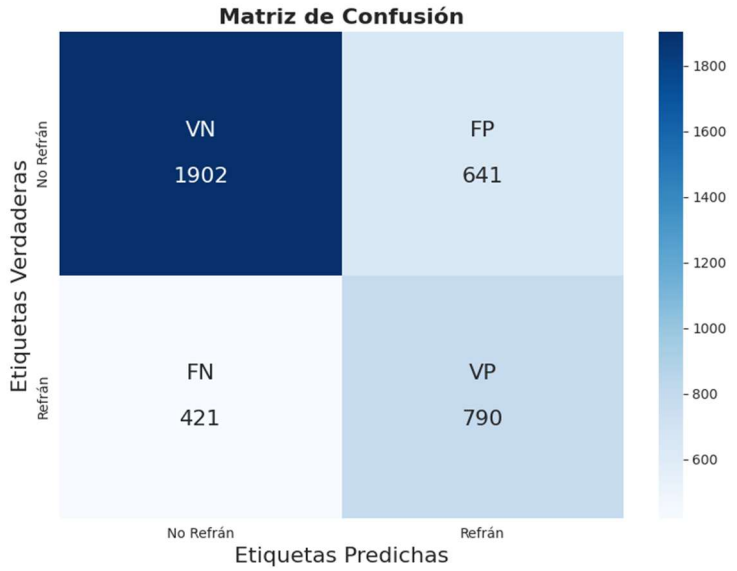

Figura 5.8 SVM con CNN (matriz de confusión)

Los resultados del informe de clasificación se presentan en la Tabla 5.14.

	Precision	Recall	F1-score	Support
No Refrán	0.82	0.75	0.78	2543
Refrán	0.55	0.65	0.60	1211
Accuracy			0.72	3754
Macro avg	0.69	0.70	0.69	3754
Weighted avg	0.73	0.72	0.72	3754

Tabla 5.14 SVM con CNN (informe de clasificación).

El modelo alcanzó un *accuracy* del 72%, lo que representa una mejora con respecto al enfoque de *Random Forest* con CNN (67%), indicando que la combinación de SVM con CNN es más efectiva en la clasificación de refranes.

La precisión para las oraciones fue de 82%, mientras que para los refranes fue de 55%, lo que sugiere que el modelo sigue presentando dificultades en la detección precisa de refranes. No obstante, el *recall* en refranes aumentó a 65%, lo que indica que el modelo logra identificar una mayor cantidad de refranes reales.

El F1-score promedio de 0.69 refleja un equilibrio entre precisión y *recall* en ambas clases. La mejora en el rendimiento general sugiere que el uso de SVM con CNN permite capturar mejor las características de los refranes,

posiblemente debido a la capacidad de SVM para manejar espacios de alta dimensión y patrones complejos en combinación con la extracción de características de la CNN.

5.3 Resultados de Modelos FLAN-T5

En esta sección se muestran los resultados de las diferentes configuraciones del modelo FLAN-T5 (*small, base y large*) utilizando distintos tamaños de corpus.

5.3.1 FLAN-T5-small

El modelo FLAN-T5-small fue evaluado en seis experimentos diferentes, como se detalla en la sección 4.7.2, utilizando corpus de tamaños incrementales y configuraciones variadas.

5.3.1.1 Resultado de los Experimentos

A continuación, se describe los resultados obtenidos para cada experimento.

Experimento 1: Corpus 2

En este experimento inicial se utiliza el Corpus 2, obteniendo un patrón de identificación desequilibrado. El modelo muestra una precisión perfecta (1.00) para identificar textos como no refranes, pero un *recall* limitado (0.57) para esta categoría. Por el contrario, para los textos que son refranes el modelo obtiene un *recall* perfecto (1.00) pero una precisión moderada (0.58). Esta tendencia sugiere que el modelo tiene una preferencia por clasificar textos como refranes.

	Precision	Recall	F1-score	Support
No Refrán	1.00	0.57	0.73	822
Refrán	0.58	1.00	0.74	496
Accuracy			0.73	1318
Macro avg	0.79	0.78	0.73	1318
Weighted avg	0.84	0.73	0.73	1318

Tabla 5.15 FLAN-T5 Small: Experimento 1 (informe de clasificación).

La Figura 5.9 muestra la matriz de confusión obtenida.

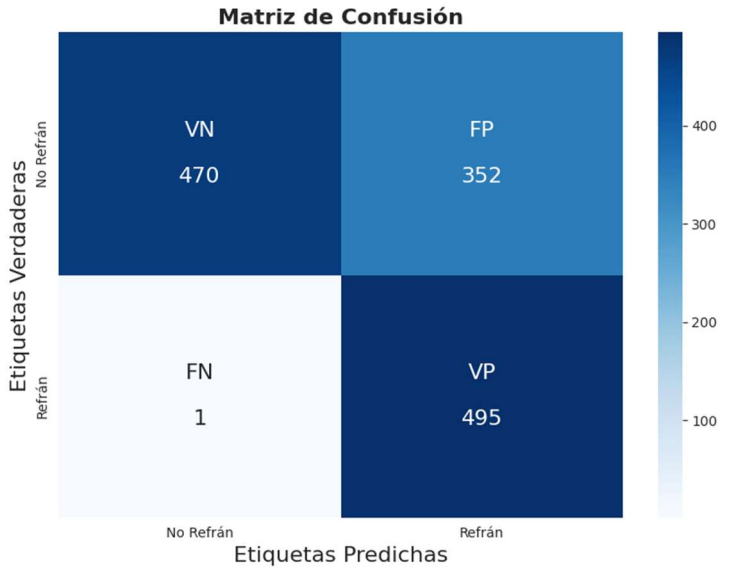

Figura 5.9 FLAN-T5 Small: Experimento 1 (matriz de confusión).

Experimento 2: Corpus 2

El segundo experimento con el mismo Corpus 2 muestra resultados casi idénticos al primer experimento, lo que confirma la consistencia del comportamiento del modelo.

	Precision	Recall	F1-score	Support
No Refrán	1.00	0.57	0.72	822
Refrán	0.58	1.00	0.73	496
Accuracy			0.73	1318
Macro avg	0.79	0.78	0.73	1318
Weighted avg	0.84	0.73	0.73	1318

Tabla 5.16 FLAN-T5 Small: Experimento 2 (informe de clasificación).

La Figura 5.10 muestra la matriz de confusión obtenida.

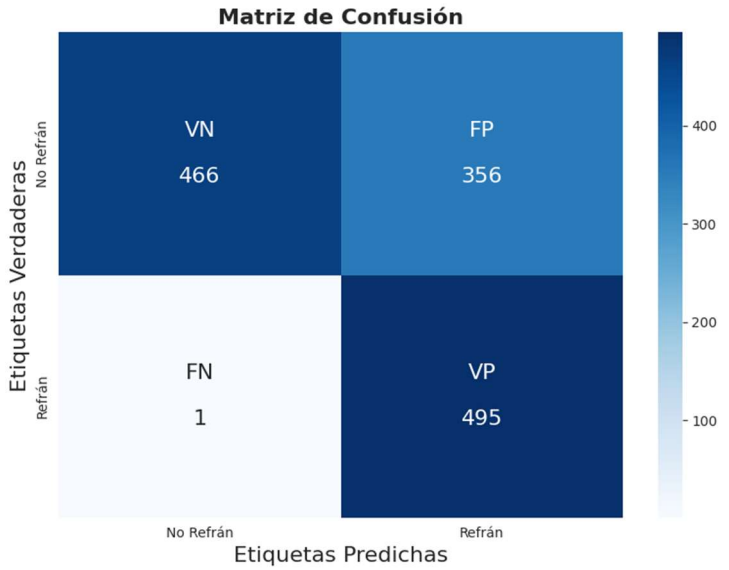

Figura 5.10 FLAN-T5 Small: Experimento 2 (matriz de confusión).

Experimento 3: Corpus 3

Al utilizar el Corpus 3 que incorpora ejemplos negativos basados en trigramas, se observa un cambio significativo en el comportamiento del modelo. Ahora presenta un sesgo extremo hacia las oraciones que no son refranes, con un *recall* casi perfecto (0.99) para esta categoría, pero un *recall* extremadamente bajo (0.02) para las oraciones que son refranes. Esta inversión del sesgo anterior sugiere que el incremento en el tamaño y la diversidad del corpus ha afectado negativamente la capacidad del modelo para reconocer refranes.

	Precision	Recall	F1-score	Support
No Refrán	0.58	0.99	0.73	1593
Refrán	0.63	0.02	0.04	1192
Accuracy			0.58	2785
Macro avg	0.60	0.51	0.39	2785
Weighted avg	0.6	0.58	0.43	2785

Tabla 5.17 FLAN-T5 Small: Experimento 3 (informe de clasificación).

La Figura 5.11 muestra la matriz de confusión obtenida.

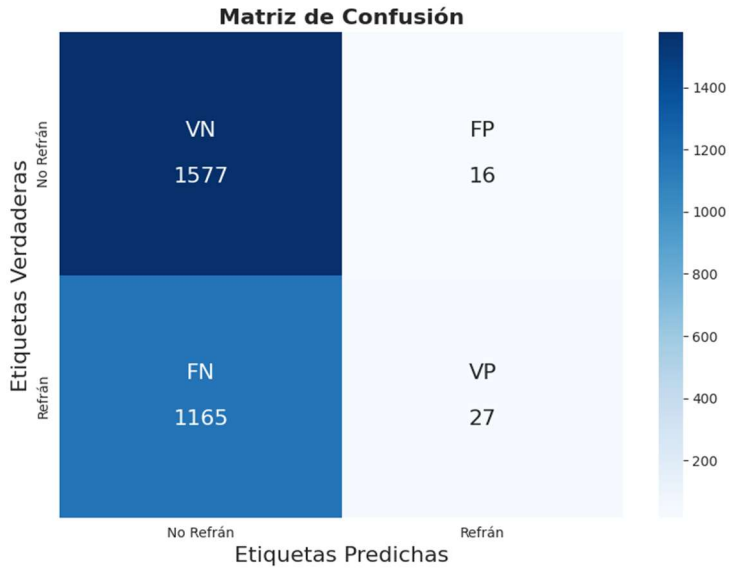

Figura 5.11 FLAN-T5 Small: Experimento 3 (matriz de confusión).

Experimento 4: Corpus 4

En este experimento se utiliza el Corpus 4 que incluye oraciones construidas a partir de los trigramas, el modelo muestra un comportamiento más equilibrado, aunque todavía presenta sesgos. La precisión para las oraciones que no son refranes es alta (0.92), pero con *recall* moderado (0.49). Para los refranes, la relación se invierte con precisión moderada (0.56) y *recall* alto (0.94). Esta configuración logra un mejor equilibrio entre las dos categorías.

	Precision	Recall	F1-score	Support
No Refrán	0.92	0.49	0.64	1772
Refrán	0.56	0.94	0.70	1219
Accuracy			0.67	2991
Macro avg	0.74	0.72	0.67	2991
Weighted avg	0.77	0.67	0.67	2991

Tabla 5.18 FLAN-T5 Small: Experimento 4 (informe de clasificación).

La Figura 5.12 muestra la matriz de confusión obtenida.

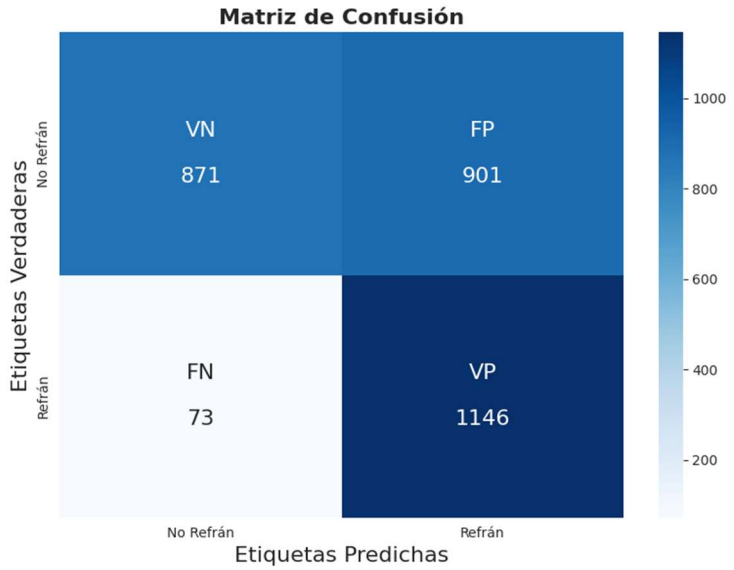

Figura 5.12 FLAN-T5 Small: Experimento 4 (matriz de confusión).

Experimento 5: Corpus 5

En el quinto experimento, utilizando el Corpus 5 (el más extenso), muestra un comportamiento similar a la observada en el Experimento 3. El modelo nuevamente presenta un fuerte sesgo hacia las oraciones que no son refranes, con *recall* casi perfecto (0.99) pero precisión moderada (0.68). El rendimiento para detectar refranes es extremadamente pobre (*recall* de 0.02), lo que sugiere que el incremento en la complejidad del corpus ha acentuado dificultades para reconocer patrones en refranes.

	Precision	Recall	F1-score	Support
No Refrán	0.68	0.99	0.80	2543
Refrán	0.35	0.02	0.03	1211
Accuracy				
Macro avg			0.67	3754
Weighted avg	0.51	0.50	0.42	3754

Tabla 5.19 FLAN-T5 Small: Experimento 5 (informe de clasificación).

La Figura 5.13 muestra la matriz de confusión obtenida.

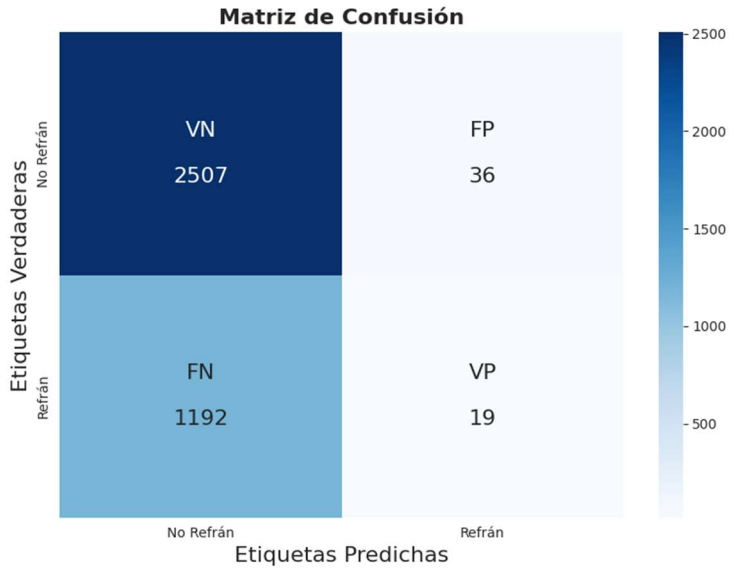

Figura 5.13 FLAN-T5 Small: Experimento 5 (matriz de confusión).

Experimento 6: Corpus 5

El experimento final representa una mejora en la identificación de refranes como se muestra en la Tabla 5.20. Este experimento utiliza el Corpus 5 juntamente con las características de rimas (número de sílabas, ritmo y palabras que riman), el modelo logra un rendimiento buen rendimiento en ambas categorías. Tanto la precisión como el *recall* son elevados para ambas clases, resultando en un F1-score alto (0.93 para las oraciones y 0.87 para los refranes) y una *accuracy* global del 91%. Este resultado demuestra que, puede aprovechar efectivamente las características lingüísticas adicionales (ritmo, número de sílabas y rimas) para lograr un rendimiento superior.

	Precision	Recall	F1-score	Support
No Refrán	0.98	0.89	0.93	2543
Refrán	0.81	0.96	0.87	1211
Accuracy			0.91	3754
Macro avg	0.89	0.92	0.90	3754
Weighted avg	0.92	0.91	0.91	3754

Tabla 5.20 FLAN-T5 Small: Experimento 6 (informe de clasificación).

La Figura 5.14 muestra la matriz de confusión obtenida.

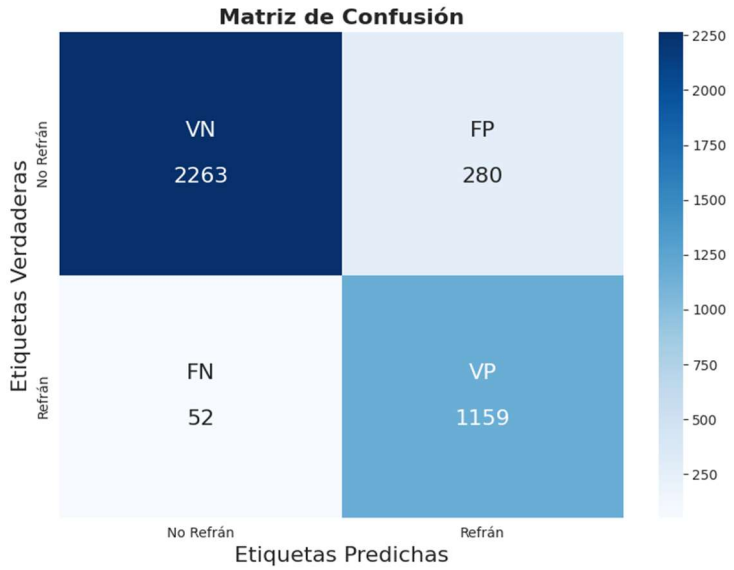

Figura 5.14 FLAN-T5 Small: Experimento 6 (matriz de confusión).

5.3.2 FLAN-T5-base

El modelo FLAN-T5-base fue evaluado en dos experimentos diferentes, utilizando el Corpus 2 para los dos experimentos y con las configuraciones detalladas en la sección 4.7.2.

5.3.2.1 Resultado de los Experimentos

A continuación, se describe los resultados obtenidos para cada experimento.

Experimento 1: Corpus 2

En este primer experimento utilizando el Corpus 2, el modelo FLAN-T5-Base muestra un comportamiento similar al modelo FLAN-T5-Small, pero con un rendimiento ligeramente superior. Como se muestra en la Tabla 5.21, el modelo presenta una precisión de 0.99 para detectar textos que no son refranes, pero con un *recall* moderado (0.62). A diferencia de los refranes que muestran un *recall* alto (0.99) con precisión moderada (0.61). Este patrón sugiere que el modelo también tiene cierta tendencia a clasificar textos como refranes, aunque mantiene mejor equilibrio que su contraparte Small, logrando una *accuracy* global del 76%.

	Precision	Recall	F1-score	Support
No Refrán	0.99	0.62	0.76	822
Refrán	0.61	0.99	0.76	496
Accuracy			0.76	1318
Macro avg	0.8	0.81	0.76	1318
Weighted avg	0.85	0.76	0.76	1318

Tabla 5.21 FLAN-T5 Base: Experimento 1 (informe de clasificación).

La Figura 5.15 muestra la matriz de confusión obtenida.

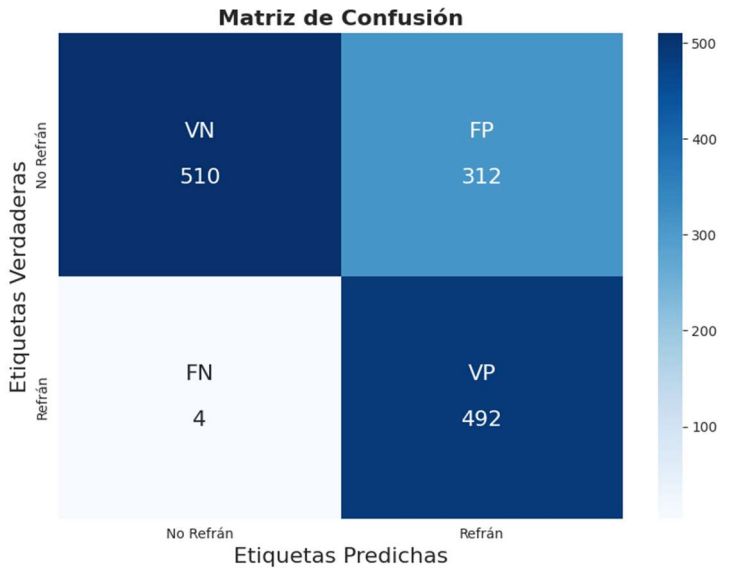

Figura 5.15 FLAN-T5 Base: Experimento 1 (matriz de confusión).

Experimento 2: Corpus 2 y con prefijo en el texto a identificar

Para el segundo experimento, se utilizó nuevamente el Corpus 2, pero incorporando un elemento adicional, es decir a cada texto a clasificar se precedió con el prefijo "clasificar: ". Esta modificación metodológica buscaba evaluar si proporcionar una instrucción explícita podría mejorar el rendimiento del modelo.

Los resultados obtenidos en la Tabla 5.22, muestran un patrón similar al experimento anterior, con una precisión alta (1.00) para las oraciones, pero con *recall* limitado (0.59), mientras que para los refranes se observa un *recall* (1.00) con precisión moderada (0.59). El *accuracy* global (74%) es ligeramente inferior al experimento sin prefijo, lo que sugiere que la adición del prefijo no aportó mejoras significativas al rendimiento del modelo.

	Precision	Recall	F1-score	Support
No Refrán	1.00	0.59	0.74	822
Refrán	0.59	1.00	0.75	496
Accuracy			0.74	1318
Macro avg	0.8	0.79	0.74	1318
Weighted avg	0.85	0.74	0.74	1318

Tabla 5.22 FLAN-T5 Base: Experimento 2 (informe de clasificación).

La Figura 5.16 muestra la matriz de confusión obtenida.

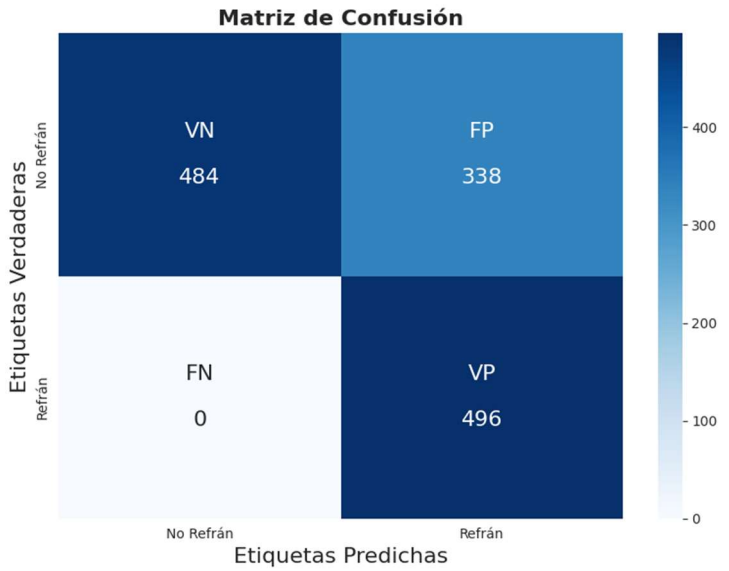

Figura 5.16 FLAN-T5 Base: Experimento 2 (matriz de confusión).

5.3.3 FLAN-T5-Large

Para obtener los resultados del modelo FLAN-T5-Large, se realizó un único experimento utilizando el Corpus 5. Este experimento permitió evaluar si un modelo de mayor capacidad podría manejar mejor la complejidad inherente a este corpus.

Los resultados obtenidos en la Tabla 5.23, muestran un rendimiento superior al observado en experimentos anteriores con modelos más pequeños. El modelo logra una precisión alta (1.00) para identificar textos que no son refranes, con un *recall* alto (0.81). Para los refranes, alcanza un *recall* alto (1.00) con una precisión buena (0.71).

Este balance entre precisión y *recall* para ambas clases se traduce en una *accuracy* global del 87%, demostrando que la mayor capacidad de FLAN-T5-Large le permite adaptarse mejor a la complejidad del Corpus 5 sin necesidad

de incorporar explícitamente características lingüísticas adicionales, como fue necesario para FLAN-T5-Small.

	Precision	Recall	F1-score	Support
No Refrán	1.00	0.81	0.89	2543
Refrán	0.71	1.00	0.83	1211
Accuracy			0.87	3754
Macro avg	0.85	0.90	0.86	3754
Weighted avg	0.91	0.87	0.87	3754

Tabla 5.23 FLAN-T5 Large: Experimento 1 (informe de clasificación).

La Figura 5.17 muestra la matriz de confusión obtenida.

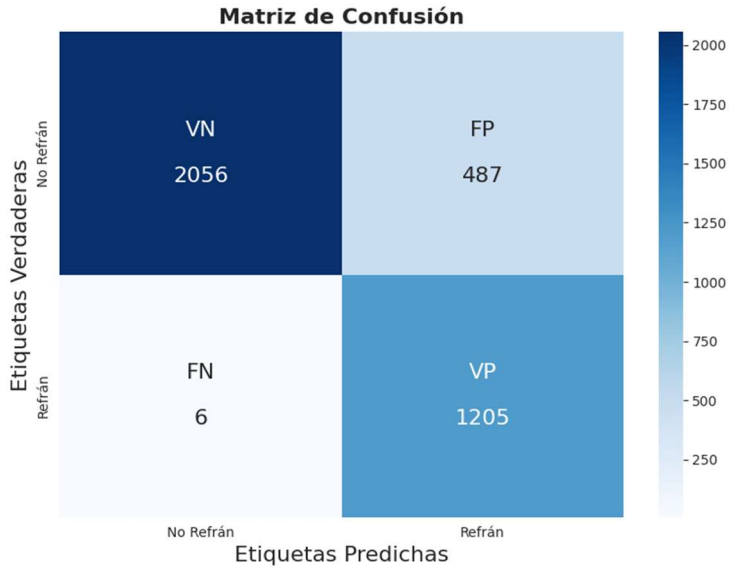

Figura 5.17 FLAN-T5 Large: Experimento 1 (matriz de confusión).

5.4 Resultados del Enfoque basado en Prompting con GPT

En este experimento se ha evaluado la eficacia de la técnica de *prompting* utilizando GPT, aplicada al Corpus 5. El modelo GPT-4o mini logra una precisión del 0.94 para las oraciones que no son refranes con un *recall* del 0.91 (ver Tabla 5.24), lo que indica una capacidad muy alta para identificar correctamente los textos que no son refranes, con pocos falsos positivos y falsos negativos. Para los refranes la precisión es de 0.83 y el *recall* de 0.89, mostrando también un rendimiento alto.

El *accuracy* global es de 0.90, lo que posiciona este enfoque entre los más efectivos de todos los evaluados en esta investigación. Este resultado es significativo considerando que se logró utilizando el corpus más complejo

(Corpus 5), lo que sugiere que los modelos basados en *prompting* pueden aprovecharse efectivamente de la riqueza y complejidad de los datos sin necesidad de entrenamientos específicos.

	Precision	Recall	F1-score	Support
No Refrán	0.94	0.91	0.93	2543
Refrán	0.83	0.89	0.85	1211
Accuracy			0.90	3754
Macro avg	0.88	0.90	0.89	3754
Weighted avg	0.91	0.90	0.90	3754

Tabla 5.24 Informe de clasificación para el enfoque basado en *Prompting* con GPT-4o mini usando el Corpus 5.

La Figura 5.18 muestra la matriz de confusión obtenida.

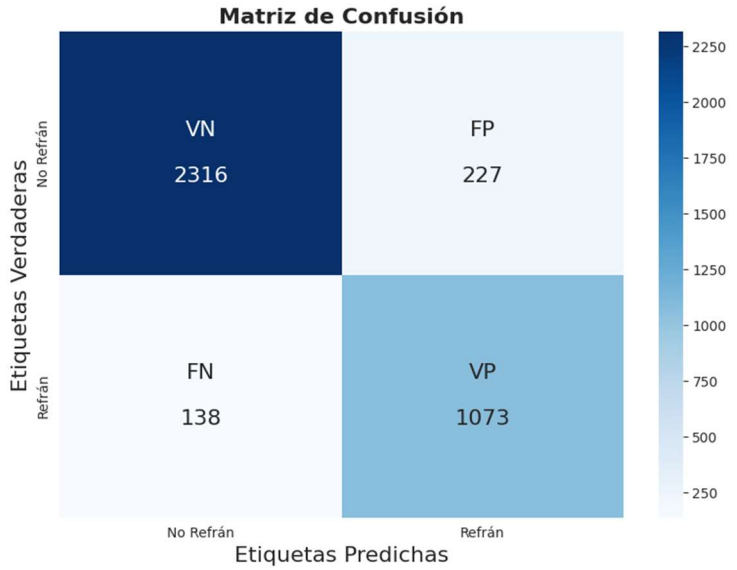

Figura 5.18 Matriz de confusión para el enfoque basado en *Prompting* con GPT-4o mini usando el Corpus 5.

5.5 Análisis Comparativo

En esta sección se realiza un análisis comparativo de los enfoques utilizados en este Trabajo de Fin de Máster, mostrando las fortalezas y debilidades de cada enfoque.

5.5.1 Análisis por Enfoque

Para cada uno de los enfoques desarrollados en este trabajo se hace un análisis de los resultados más relevantes y se termina en un análisis global que sintetiza los resultados obtenidos de todos los enfoques utilizados en este trabajo.

5.5.1.1 Clasificadores tradicionales

En este enfoque se utiliza clasificadores tradicionales (SVM, *Random Forest* y regresión logística) mostrando un comportamiento similar y con poca capacidad de generalizar, con un *accuracy* que va del 29 al 54%, esto se debe a que está aprendiendo como principal patrón el número de palabras que componen el texto para determinar cómo refrán o no refrán, ocasionando numerosos falsos positivos. Otro aspecto que considerar es el uso de lematización ya que al no lematizar el texto a identificar se obtuvo un mejor *acurracy* de alrededor del 54%, esto debido a que se mantiene las características de rima de los refranes.

5.5.1.2 Enfoques Híbridos

El incorporar una red convolucional (CNN) para obtener características profundas del texto y juntamente con el uso de FastText que maneja las palabras fuera del vocabulario mejoró significativamente el rendimiento, teniendo un *accuracy* de hasta un 73% en el mejor caso (regresión logística con CNN), además este enfoque tiene un mejor equilibrio entre precisión y *recall* para la identificación de textos y refranes, reduciendo el sesgo observado en los clasificadores tradicionales.

5.5.1.3 Modelos FLAN-T5

En lo que corresponde a los modelos FLAN-T5 se tiene resultados variables dependiendo del tamaño del modelo y el corpus utilizado.

- FLAN-T5 Small: Con los corpus más pequeños (Corpus 2, 3 y 4) se tiene resultados similares al enfoque con clasificadores tradicionales, mientras que utilizando el Corpus 5 y características lingüísticas adicionales (rima) alcanza el rendimiento más alto (91%).
- FLAN-T5-Base: Mejora ligeramente el rendimiento de FLAN-T5-Small en el Corpus 2 (76%).
- FLAN-T5-Large: Logra un excelente rendimiento (87%) con el Corpus 5, sin necesidad de características lingüísticas adicionales.

5.5.1.4 Enfoque Basado en Prompting con GPT

El enfoque basado en *prompting* con GPT muestra un rendimiento alto (90%), comparable al mejor resultado obtenido con FLAN-T5-Small y mostrando un mejor balance en la identificación de refranes y textos que no son refranes. La ventaja principal de este enfoque es que logra este alto rendimiento sin

necesidad de un entrenamiento específico, aprovechando el conocimiento previo del modelo.

5.5.2 Análisis Comparativo Global

Tras evaluar diversos enfoques para la identificación de refranes, en esta sección se presenta un análisis comparativo global que permite identificar las fortalezas, limitaciones y eficacia relativa de cada método evaluado.

La Tabla 5.25 sintetiza los mejores resultados obtenidos con cada enfoque evaluado en este Trabajo de Fin de Máster, incluyendo las métricas de *accuracy*, *precision*, *recall* y *F1-score* tanto por clase como en promedio.

Enfoque	Corpus	Accuracy	Precision (NR/R)	Recall (NR/R)	F1-score (NR/R)	Macro Avg F1
Regresión logística tradicional	Corpus 2	0.54	0.38/1.00	1.00/0.36	0.56/0.53	0.54
Regresión logística + CNN	Corpus 5 (con características de rima)	0.73	0.90/0.55	0.67/0.85	0.77/0.67	0.72
SVM + CNN	Corpus 5	0.72	0.82/0.55	0.75/0.65	0.78/0.60	0.69
FLAN-T5 Small	Corpus 5 (con características de rima)	0.91	0.98/0.81	0.89/0.96	0.93/0.87	0.90
FLAN-T5 Base	Corpus 2	0.76	0.99/0.61	0.62/0.99	0.76/0.76	0.76
FLAN-T5 Large	Corpus 5	0.87	1.00/0.71	0.81/1.00	0.89/0.83	0.86
GPT-4o mini con prompting	Corpus 5	0.90	0.94/0.83	0.91/0.89	0.93/0.85	0.89

Tabla 5.25 Comparación de métricas de rendimiento entre los diferentes enfoques evaluados (NR = No Refrán, R = Refrán).

5.6 Análisis del Comportamiento en Contexto de Producción

En esta sección se presenta el comportamiento que tienen los modelos que lograron una mejor puntuación, pero en contextos de producción real. Revelando hallazgos significativos para implementaciones prácticas.

5.6.1 Brecha entre Evaluación Controlada y Contexto Real

Para la evaluación del desempeño dentro de contexto real, se utilizó un conjunto independiente de datos de producción descrito en la Sección 5 (Tabla 5.4), compuesto por 9 refranes y 26 oraciones que no son refranes (35 oraciones en total) con el objetivo de verificar el comportamiento de los modelos en este escenario. Este conjunto representa textos que aparecen naturalmente en

contextos reales, permitiendo verificar la robustez de los modelos fuera de las condiciones controladas de laboratorio, teniendo como resultado la Tabla 5.26, que compara el *accuracy* obtenido con los datos de prueba y los de producción para los enfoques donde su *accuracy* en prueba fueron significativamente altos.

Enfoque	Accuracy con datos de prueba	Accuracy con datos de producción
Regresión logística + CNN	0.73	0.31
FLAN-T5 Small	0.91	0.49
FLAN-T5 Large	0.87	0.34
GPT-4o mini con prompting	0.90	0.90

Tabla 5.26 Comparación del rendimiento entre datos de prueba controlados y datos de producción real.

Los resultados revelan una gran diferencia entre el rendimiento en condiciones de evaluación controlada y el observado en un contexto de producción real para casi todos los modelos evaluados:

- Regresión logística + CNN: Muestra una reducción de 57.5%, con su *accuracy* cayendo de 0.73 a 0.31 en producción.
- FLAN-T5 Small: A pesar de lograr la mejor puntuación en evaluación controlada (0.91), sufre una degradación del 46.2%, reduciendo su accuracy a 0.49 en producción.
- FLAN-T5 Large: Experimenta la reducción más severa (60.9%), pasando de un *accuracy* 0.87 a solo 0.34 en producción.
- GPT-4o mini con *prompting*: Mantiene exactamente el mismo *accuracy* (0.90) tanto en evaluación controlada como en producción, demostrando una robustez sobresaliente.

5.6.2 Análisis de las Causas de Degradación

La marcada degradación observada en la mayoría de los modelos puede atribuirse a varios factores relacionados tanto con las características de los refranes como con las limitaciones de los conjuntos de datos utilizados:

5.6.2.1 Desafíos relacionados con las características lingüísticas

Si bien los refranes están fuertemente relacionados al contexto cultural de una región y frecuentemente se caracterizan por presentar rimas [60], esta característica puede causar que los modelos entrenados muestren dificultad

para discriminar entre refranes genuinos y otros textos que contienen estructuras rítmicas o rimas similares.

FLAN-T5 Small, a pesar de utilizar características lingüísticas como rimas y patrones rítmicos, muestra una alta sensibilidad a variaciones en estas características cuando se presentan en textos reales. Esta sensibilidad se convierte en una limitación al encontrarse con textos que, sin ser refranes, presentan rimas y estructuras de verso similares [60]. En el conjunto de datos de producción, la inclusión deliberada de fragmentos de poemas que contienen rimas resultó ser particularmente problemática, causando numerosas clasificaciones incorrectas.

5.6.2.2 Limitaciones en la representatividad del corpus de entrenamiento

Otro factor que se ha identificado es la variabilidad limitada del conjunto de entrenamiento. Si bien los textos que no son refranes fueron extraídos del corpus AnCora, muchos de estos ejemplos no presentan estructuras lingüísticas similares a los refranes. Esta diferencia estructural entre las clases facilita la discriminación durante el entrenamiento, pero no prepara adecuadamente a los modelos para enfrentar casos más desafiantes en producción.

Los modelos no aprenden a identificar las características principales que distinguen a un refrán más allá de patrones superficiales, sino que desarrollan heurísticas simplificadas basadas en las diferencias más evidentes entre las clases en el conjunto de entrenamiento o en muchos casos en lugar de entender sus características esenciales, se fijan en pistas superficiales como palabras comunes que aparecen en muchos refranes (como "dios", "ayuda", "mal" o "bien"). Cuando estas heurísticas se enfrentan a textos más variados y complejos en producción, su rendimiento se degrada significativamente.

5.6.2.3 Complejidad contextual y variabilidad en producción

Los textos en producción real presentan mayor variabilidad lingüística, diversidad estructural y complejidad contextual que no están suficientemente representadas en los conjuntos de entrenamiento y prueba controlados. Siendo estos:

- Variantes regionales o temporales de refranes conocidos
- Refranes modificados o adaptados al contexto
- Expresiones en la frontera entre refranes y otros géneros (como proverbios, dichos populares, o citas)
- Textos literarios con estructuras similares a refranes (como versos de poemas o letras de canciones)

En contraste con estos análisis, se observa la robustez de GPT-4o mini con *prompting* que parece derivar de su capacidad para integrar no solo características estructurales y lingüísticas, sino también aspectos semánticos y

culturales asociados a los refranes, producto de su exposición a un corpus masivo y diverso durante su preentrenamiento.

5.7 Análisis de Casos Específicos de Clasificación

En esta sección se complementa el análisis cuantitativo presentando, donde se examina casos específicos de clasificación que ilustran las limitaciones de los modelos evaluados. Con el objetivo de comprender mejor las características que facilitan o dificultan la correcta identificación de refranes.

5.7.1 Falsos Negativos: Refranes Clasificados como No Refranes

En Tabla 5.27 se muestran ejemplos significativos de clasificaciones donde el modelo clasifica incorrectamente refranes como textos ordinarios. Pocos fueron los casos de falsos negativos, indicando que todos los modelos frecuentemente fallan al clasificar textos como refranes (falsos positivos)

Texto	Modelo	Clasificación Correcta	Clasificación Obtenida
“En casa de informático, ordenador lleno de virus”	Regresión logística + CNN	Refrán	No refrán
“Escoba nueva barre bien”	FLAN-T5 Large	Refrán	No refrán

Tabla 5.27 Falsos Negativos: Refranes Clasificados como No Refranes

Estos falsos negativos revelan limitaciones en los modelos siendo estos:

"**En casa de informático, ordenador lleno de virus**" representa una adaptación moderna del refrán tradicional "En casa de herrero, cuchillo de palo". La incapacidad del modelo de regresión logística con CNN para reconocerlo sugiere una dificultad para identificar variantes contemporáneas que mantienen la estructura y función de refranes tradicionales, pero incorporan léxico moderno ("informático", "ordenador", "virus"). Este error muestra que el modelo no ha aprendido la estructural del refrán, sino que posiblemente depende de vocabulario específico asociado a refranes tradicionales.

"**Escoba nueva barre bien**", a pesar de ser un refrán tradicional con estructura típica, no fue reconocido por FLAN-T5 Large. Este caso es particularmente ya que este refrán contiene características propias de un refrán como brevedad, mensaje metafórico y aplicabilidad general [55], [57]. Su no detección sugiere que el modelo puede estar condicionado por características más superficiales como la presencia de ciertas palabras clave o estructuras sintácticas específicas.

5.7.2 Falsos Positivos: No Refranes Clasificados como Refranes

En la Tabla 5.28 se muestra los casos en donde el modelo clasifica un texto ordinario como refrán, en este caso todos los modelos tuvieron dificultad.

Texto	Modelo	Clasificación Correcta	Clasificación Obtenida
“A la piedra piedra, y al zapato la piedra”	Regresión logística + CNN	No refrán	Refrán
“verde, amarilla, gris, blanca en la cumbre”	FLAN-T5 Small	No refrán	Refrán
“Juan de Dios salió a trabajar en la madrugada”	FLAN-T5 Large	No refrán	Refrán
“Al pan con mantequilla, más mantequilla todavía”	GPT-4o mini con prompting	No refrán	Refrán
“He vuelto a creer en Dios”	Regresión logística + CNN	No refrán	Refrán
“A veces voy por un camino”	FLAN-T5 Small	No refrán	Refrán
“Él tomó vino y comió pan”	FLAN-T5 Large	No refrán	Refrán
“A la piedra piedra, y al zapato la piedra”	GPT-4o mini con prompting	No refrán	Refrán

Tabla 5.28 Falsos Positivos: No Refranes Clasificados como Refranes.

Al analizar estos falsos positivos se encontraron las siguientes características que hacen que un modelo tenga una clasificación errónea.

5.7.2.1 Imitación de estructuras sintácticas típicas de refranes

"A la piedra piedra, y al zapato la piedra" y *"Al pan con mantequilla, más mantequilla todavía"* imitan la estructura bimembre y la formulación sintáctica característica de muchos refranes [56], y empleando paralelismos o repeticiones en sus palabras. Este error es significativo que incluso GPT-4o mini, el modelo más robusto en contextos de producción, cometa errores con estas construcciones que emulan estructuralmente a refranes.

5.7.2.2 Presencia de léxico frecuentemente asociado a refranes

"He vuelto a creer en Dios", "Juan de Dios salió a trabajar en la madrugada" y *"Él tomó vino y comió pan"* contienen palabras frecuentemente presentes en refranes tradicionales (como "Dios", "pan", "vino"). Esto confirma que los modelos dan mayor peso a la presencia de cierto vocabulario, independientemente del contexto y función del texto.

5.7.2.3 Características rítmicas o poéticas

"verde, amarilla, gris, blanca en la cumbre" presenta una estructura rítmica que podría recordar a características poéticas presentes en algunos refranes.

"A veces voy por un camino" tiene una cadencia y brevedad que podría evocar el inicio de un refrán o una expresión sentenciosa.

6 Conclusiones y Líneas Futuras

El presente Trabajo de Fin de Máster ha abordado el análisis de diferentes técnicas subsimbólicas para la de la identificación de refranes en textos escritos en español mediante diversos enfoques. A través de un análisis comparativo, se han evaluado métodos que van desde clasificadores tradicionales de aprendizaje automático hasta modelos de lenguaje de gran escala. Los resultados obtenidos permiten extraer conclusiones relevantes tanto a nivel técnico como lingüístico.

En primer lugar, se ha observado que la efectividad de los modelos está fuertemente relacionada con las características lingüísticas incorporadas en su entrenamiento. Los experimentos demuestran que las características relacionadas con la rima y la estructura rítmica juegan un papel fundamental en la identificación de refranes, como muestra el incremento del rendimiento del modelo FLAN-T5 Small al incorporar estas características, alcanzando un 91% de precisión en entornos controlados. Por el contrario, la lematización afecta negativamente la identificación al eliminar componentes rítmicos esenciales que diferencian a los refranes de otras expresiones lingüísticas.

Un hallazgo significativo es la diferencia entre el rendimiento en entornos controlados y en contextos de producción real. Mientras que varios modelos mostraron resultados prometedores en evaluaciones de laboratorio, solo el enfoque basado en GPT-4o mini con *prompting* mantuvo su rendimiento (90%) en ambos escenarios. Los demás modelos experimentaron degradaciones en su rendimiento: Regresión logística con CNN descendió de 73% a 31%, FLAN-T5 Small de 91% a 49% y FLAN-T5 Large de 87% a 34%.

En el análisis cualitativo de los casos específicos de clasificación ha permitido identificar patrones consistentes en los errores cometidos por los modelos. Los falsos negativos (refranes no reconocidos) ocurren frecuentemente con refranes modernos o adaptaciones de estos, mientras que los falsos positivos (textos ordinarios clasificados como refranes) aparecen cuando los textos imitan estructuras sintácticas típicas de refranes o contienen léxico frecuentemente asociado a expresiones paremiológicas. Esto revela que los modelos desarrollan heurísticas basadas en características superficiales en lugar de capturar la esencia de los refranes como expresiones culturales.

La comparación entre diferentes arquitecturas también aporta conclusiones importantes sobre sus capacidades y limitaciones. Los clasificadores tradicionales, aunque computacionalmente eficientes, muestran un rendimiento limitado y alta sensibilidad a variaciones lingüísticas. La incorporación de redes convolucionales mejora significativamente la capacidad de extracción de características, pero sigue presentando problemas de generalización. Los modelos preentrenados como FLAN-T5 logran excelentes resultados cuando se complementan con características específicas, pero su rendimiento se reduce considerablemente en contextos reales. Finalmente, el enfoque basado en *prompting* con GPT-4o mini demuestra una robustez, sugiriendo que el conocimiento lingüístico generalizado adquirido durante el

preentrenamiento a gran escala proporciona una base más sólida para esta tarea que el entrenamiento específico en conjuntos de datos limitados.

Estos resultados tienen implicaciones importantes para el desarrollo futuro de sistemas de identificación de refranes en textos escritos en español. Sugieren que una aproximación debería combinar el conocimiento generalizado de modelos preentrenados con características lingüísticas específicas de los refranes, especialmente aquellas relacionadas con su estructura rítmica y su función en el contexto comunicativo y social. Asimismo, se evidencia la necesidad de desarrollar conjuntos de entrenamiento más diversos que incluyan ejemplos negativos más variados y que capturen características similares a los refranes, como primer enfoque en este trabajo se utilizó diferentes corpus para el entrenamiento de los modelos, pero esto no es suficiente, para forzar a los modelos a desarrollar representaciones más profundas. Se puede considerar crear un corpus con refranes auténticos, textos que parezcan refranes pero que no lo son y conjuntamente con variantes modernas de refranes tradicionales que sería un enfoque similar al utilizado para la probar la interpretación del lenguaje figurado realizado por Liu, Cui, Zheng y Neubig (2022) [1].

En conclusión, la identificación de refranes va más allá del reconocimiento de patrones lingüísticos superficiales, requiriendo una comprensión más profunda del contexto cultural y la función de estas expresiones. Los modelos más efectivos son aquellos que logran capturar estas dimensiones más allá de características léxicas. El enfoque basado en GPT-4o mini con *prompting* emerge como la solución más robusta actualmente, aunque futuros avances podrían surgir de la integración más sofisticada de conocimiento lingüístico específico en arquitecturas de aprendizaje profundo.

7 Bibliografía

[1] E. Liu, C. Cui, K. Zheng, y G. Neubig, «Testing the Ability of Language Models to Interpret Figurative Language», 15 de mayo de 2022, *arXiv*: arXiv:2204.12632. Accedido: 5 de agosto de 2024. [En línea]. Disponible en: http://arxiv.org/abs/2204.12632

[2] T. Kalandadze, C. Norbury, T. Nærland, y K.-A. B. Næss, «Figurative language comprehension in individuals with autism spectrum disorder: A meta-analytic review», *Autism*, vol. 22, n.º 2, pp. 99-117, feb. 2018, doi: 10.1177/1362361316668652.

[3] L. A. F. Uribe, «ACERCAMIENTO TEÓRICO A LA COMPRENSIÓN-INTERPRETACIÓN DEL LENGUAJE FIGURADO EN APRENDICES TARDÍOS DE L2», vol. 26.

[4] D. Bogdanova, «A Framework for Figurative Language Detection Based on Sense Differentiation».

[5] V. Sammartino, C. Baccheschi, J. Gneri, V. Picchianti, y A. Frasson, «Feature Engineering and Semantic Enrichment for Enhanced Text Classification: A Case Study on Figurative Language in Tweets», 13 de agosto de 2024. doi: 10.21203/rs.3.rs-4783494/v1.

[6] T. Zhang, C. Li, N. Cao, R. Ma, S. Zhang, y N. Ma, «Text Feature Extraction and Classification Based on Convolutional Neural Network (CNN)», en *Data Science*, vol. 727, B. Zou, M. Li, H. Wang, X. Song, W. Xie, y Z. Lu, Eds., en Communications in Computer and Information Science, vol. 727. , Singapore: Springer Singapore, 2017, pp. 472-485. doi: 10.1007/978-981-10-6385-5_40.

[7] S. B. Razali, «Figurative Language Detection using Deep Learning and Contextual Features».

[8] N. R. Norrick, «1 Subject Area, Terminology, Proverb Definitions, Proverb Features», en *Introduction to Paremiology: A Comprehensive Guide to Proverb Studies*, H. Hrisztova-Gotthardt y M. Aleksa Varga, Eds., Warsaw, Poland: DE GRUYTER OPEN, 2015. doi: 10.2478/9783110410167.1.

[9] E. V. Shutova, «Computational approaches to figurative language».

[10] D. M. Colán, «El papel de los refranes en la compresión y producción del texto».

[11] M. Koszla-Szvmanka, «Los proverbios y refranes en la enseñanza de la lengua española».

[12] T. Kalandadze, «Comprensión lectora basada en evidencias». [En línea]. Disponible en: https://clbe.wordpress.com/tag/lenguaje-figurado/

[13] M. Á. Idiazábal Alecha y E. Boque Hermida, «Procesamiento cognitivo en los trastornos del espectro autista», *RevNeurol*, vol. 44, n.º S02, p. S049, 2007, doi: 10.33588/rn.44S02.2006659.

[14] P. F. P. Rivero y L. M. M. Garrido, «Perfiles cognitivos en el Trastorno Autista de Alto Funcionamiento y el Síndrome de Asperger», 2014.

[15] A. Araujo, «Lenguaje en Autismo: Desafíos y Características Clave». [En línea]. Disponible en: https://mentaltestlab.com/lenguaje-en-autismo-desafios-y-caracteristicas-clave/

[16] S. G. Rodríguez y J. U. Rodríguez, «Modelado Conceptual de un Algoritmo de Aprendizaje de Máquina para la Identificación Automática de Metaforas Conceptuales a partir de Revisión de Literatura».

[17] «¿Qué es el PLN (procesamiento del lenguaje natural)?» [En línea]. Disponible en: https://www.ibm.com/es-es/topics/natural-language-processing

[18] C. Raffel *et al.*, «Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer», 19 de septiembre de 2023, *arXiv*: arXiv:1910.10683. Accedido: 29 de septiembre de 2024. [En línea]. Disponible en: http://arxiv.org/abs/1910.10683
[19] «Procesamiento del Lenguaje Natural (PNL): Conceptos y aplicaciones». [En línea]. Disponible en: https://www.isdi.education/es/blog/procesamiento-del-lenguaje-natural
[20] M. Hernandez y Gómez, «Aplicaciones de Procesamiento de Lenguaje Natural», *Revista Politécnica*, vol. 32, pp. 87-96, jul. 2013.
[21] J. P. M. Montiel, «Detección y corrección de errores basados en reglas gramaticales del inglés en conjunto con el mejoramiento de estilos de escritura, aplicado en artículos científicos mediante Algoritmos de Procesamiento de Lenguaje Natural (NLP)».
[22] D. Moreira *et al.*, «Análisis del Estado Actual de Procesamiento de Lenguaje Natural».
[23] Koroteev, «BERT: A Review of Applications in Natural Language Processing and Understanding», 2021, [En línea]. Disponible en: https://arxiv.org/abs/2103.11943
[24] N. Muennighoff, «SGPT: GPT Sentence Embeddings for Semantic Search», 5 de agosto de 2022, *arXiv*: arXiv:2202.08904. Accedido: 5 de junio de 2024. [En línea]. Disponible en: http://arxiv.org/abs/2202.08904
[25] M. L. N. Brotons, «Las paremias y sus variantes: análisis sintáctico, semántico y traductológico español/francés.».
[26] M. A. Hearst, «Automatic Acquisition of Hyponyms from Large Text Corpora», vol. COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics, 1992, [En línea]. Disponible en: https://aclanthology.org/C92-2082/
[27] E. M. Bender y A. Koller, «Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data», en *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 5185-5198. doi: 10.18653/v1/2020.acl-main.463.
[28] A. P. Rassi, J. Baptista, y O. Vale, «Automatic Detection of Proverbs and their Variants», *OASIcs, Volume 38, SLATE 2014*, vol. 38, pp. 235-249, 2014, doi: 10.4230/OASICS.SLATE.2014.235.
[29] *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition*. Accedido: 13 de enero de 2025. [En línea]. Disponible en: https://learning.oreilly.com/library/view/hands-on-machine-learning/9781098125967/
[30] M. del Pilar Salas-Zárate, G. Alor-Hernández, J. L. Sánchez-Cervantes, M. A. Paredes-Valverde, J. L. García-Alcaraz, y R. Valencia-García, «Review of English literature on figurative language applied to social networks», *Knowl Inf Syst*, vol. 62, n.º 6, pp. 2105-2137, jun. 2020, doi: 10.1007/s10115-019-01425-3.
[31] E. Sulis, D. Irazú Hernández Farías, P. Rosso, V. Patti, y G. Ruffo, «Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not», *Knowledge-Based Systems*, vol. 108, pp. 132-143, sep. 2016, doi: 10.1016/j.knosys.2016.05.035.
[32] «What Is Random Forest? | IBM». Accedido: 13 de enero de 2025. [En línea]. Disponible en: https://www.ibm.com/think/topics/random-forest
[33] Y. Izza, A. Ignatiev, y J. Marques-Silva, «On Explaining Decision Trees», 21 de octubre de 2020, *arXiv*: arXiv:2010.11034. doi: 10.48550/arXiv.2010.11034.

[34] T. Mikolov, K. Chen, G. Corrado, y J. Dean, «Efficient Estimation of Word Representations in Vector Space», 7 de septiembre de 2013, *arXiv*: arXiv:1301.3781. doi: 10.48550/arXiv.1301.3781.
[35] J. Devlin, M.-W. Chang, K. Lee, y K. Toutanova, «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding», 24 de mayo de 2019, *arXiv*: arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805.
[36] A. Vaswani *et al.*, «Attention Is All You Need», 1 de agosto de 2023, *arXiv*: arXiv:1706.03762. Accedido: 4 de junio de 2024. [En línea]. Disponible en: http://arxiv.org/abs/1706.03762
[37] H. W. Chung *et al.*, «Scaling Instruction-Finetuned Language Models», 6 de diciembre de 2022, *arXiv*: arXiv:2210.11416. doi: 10.48550/arXiv.2210.11416.
[38] E. Cambria, «Affective Computing and Sentiment Analysis», *IEEE INTELLIGENT SYSTEMS*, 2016.
[39] S. Ruder, I. Vulić, y A. Søgaard, «A Survey of Cross-lingual Word Embedding Models», *jair*, vol. 65, pp. 569-631, ago. 2019, doi: 10.1613/jair.1.11640.
[40] «Proverbia». [En línea]. Disponible en: https://proverbia.net/
[41] «Centro Virtual Cervantes». [En línea]. Disponible en: https://cvc.cervantes.es/
[42] «Open Multilingual Wordnet». [En línea]. Disponible en: https://omwn.org/
[43] J. Sevilla y E. Cases Berbel, *Refranes del siglo XVI en el siglo XXI*. en Supplement series of Proverbium, no. volume 44. Burlington, Vermont: «Proverbium» ; University of Vermont, 2020.
[44] O. Tarnovska, «Diagnóstico del refrán en el español actual.».
[45] X. A. Álvarez Pérez, «Distribución geográfica de los refranes. Notas para el análisis geoparemiológico», *AFEL*, vol. 5, pp. 25-52, dic. 2015, doi: 10.1344/AFEL.2015.5.3.
[46] M. Grandini, E. Bagli, y G. Visani, «Metrics for Multi-Class Classification: an Overview», 13 de agosto de 2020, *arXiv*: arXiv:2008.05756. doi: 10.48550/arXiv.2008.05756.
[47] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, y Y. Artzi, «BERTScore: Evaluating Text Generation with BERT», 24 de febrero de 2020, *arXiv*: arXiv:1904.09675. doi: 10.48550/arXiv.1904.09675.
[48] W. Xu, C. Napoles, E. Pavlick, Q. Chen, y C. Callison-Burch, «Optimizing Statistical Machine Translation for Text Simplification», *TACL*, vol. 4, pp. 401-415, dic. 2016, doi: 10.1162/tacl_a_00107.
[49] C.-Y. Lin, «ROUGE: A Package for Automatic Evaluation of Summaries».
[50] B. Bernar, H. Winters, L. Fischer, B. Meyer, y M. Gyllenborg, «Exploring the Concept of Dynamic Memory Persistence in Large Language Models for Optimized Contextual Comprehension», 7 de noviembre de 2024. doi: 10.22541/au.173101368.88048313/v1.
[51] J. P. Kincaid, Jr. Fishburne, R. Robert P., C. Richard L., y Brad S., «Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel»:, Defense Technical Information Center, Fort Belvoir, VA, feb. 1975. doi: 10.21236/ADA006655.
[52] J. M. Sbarbi y Osuna, *Diccionario de refranes, adagios, proverbios, modismos, locuciones y frases proverbiales*.
[53] «CVC. Refranero multilingüe. Lista alfabética de paremias (A).» Accedido: 15 de diciembre de 2024. [En línea]. Disponible en: https://cvc.cervantes.es/lengua/refranero/listado.aspx

[54] M. Taule, M. A. Martı, y M. Recasens, «AnCora: Multilevel Annotated Corpora for Catalan and Spanish».
[55] L. A. Hernando Cuadrado, «Estilística del refrán», *Paremia*, n.º 6, pp. 327-332, 1997.
[56] J.-C. Anscombre, «Estructura métrica y función semántica de los refranes».
[57] J. Casares, «La frase proverbial y el refrán», *Revista institucional | UPB*, vol. 27, n.º 95, Art. n.º 95, 1964.
[58] M. Mac Coinnigh, «5 Structural Aspects of Proverbs», en *Introduction to Paremiology: A Comprehensive Guide to Proverb Studies*, H. Hrisztova-Gotthardt y M. Aleksa Varga, Eds., Warsaw, Poland: DE GRUYTER OPEN, 2015. doi: 10.2478/9783110410167.5.
[59] A. Feldman y J. Peng, «An approach to automatic figurative language detection: A pilot study».
[60] F. Barbieri, F. Ronzano, y H. Saggion, «Is This Tweet Satirical? A Computational Approach for Satire Detection in Spanish».
[61] «Google Books Ngram Viewer». Accedido: 8 de enero de 2025. [En línea]. Disponible en: https://books.google.com/ngrams/graph?content=amistad,dios,trabajo,madrugar,a&year_start=2020&year_end=2021&corpus=es&smoothing=3
[62] «API « SenticNet». Accedido: 9 de enero de 2025. [En línea]. Disponible en: https://sentic.net/api/
[63] K. Perifanos, E. Florou, y D. Goutsos, «Deep Learning based, end-to-end metaphor detection in Greek language with Recurrent and Convolutional Neural Networks», 23 de julio de 2020, *arXiv*: arXiv:2007.11949. doi: 10.48550/arXiv.2007.11949.
[64] N. T. Vu, H. Adel, P. Gupta, y H. Schütze, «Combining Recurrent and Convolutional Neural Networks for Relation Classification», 24 de mayo de 2016, *arXiv*: arXiv:1605.07333. doi: 10.48550/arXiv.1605.07333.
[65] C. Camacho, «CNNs for Text Classification». Accedido: 12 de enero de 2025. [En línea]. Disponible en: https://cezannec.github.io/CNN_Text_Classification/
[66] Y. Kim, «Convolutional Neural Networks for Sentence Classification», 3 de septiembre de 2014, *arXiv*: arXiv:1408.5882. doi: 10.48550/arXiv.1408.5882.
[67] S. Poria, E. Cambria, D. Hazarika, y P. Vij, «A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks», 27 de julio de 2017, *arXiv*: arXiv:1610.08815. doi: 10.48550/arXiv.1610.08815.
[68] M. Abulaish, A. Kamal, y M. J. Zaki, «A Survey of Figurative Language and Its Computational Detection in Online Social Networks», *ACM Trans. Web*, vol. 14, n.º 1, pp. 1-52, feb. 2020, doi: 10.1145/3375547.
[69] A. Joshi, P. Bhattacharyya, y M. J. Carman, «Automatic Sarcasm Detection: A Survey», 20 de septiembre de 2016, *arXiv*: arXiv:1602.03426. doi: 10.48550/arXiv.1602.03426.
[70] P. Bojanowski, E. Grave, A. Joulin, y T. Mikolov, «Enriching Word Vectors with Subword Information», 19 de junio de 2017, *arXiv*: arXiv:1607.04606. doi: 10.48550/arXiv.1607.04606.
[71] D. W. H. Jr, S. Lemeshow, y R. X. Sturdivant, *Applied Logistic Regression*. John Wiley & Sons, 2013.
[72] Y. Tsvetkov, L. Boytsov, A. Gershman, E. Nyberg, y C. Dyer, «Metaphor Detection with Cross-Lingual Model Transfer», en *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 248-258. doi: 10.3115/v1/P14-1024.

[73] J. Briskilal y C. N. Subalalitha, «Classification of Idioms and Literals Using Support Vector Machine and Naïve Bayes Classifier», en *Machine Vision and Augmented Intelligence—Theory and Applications*, M. K. Bajpai, K. Kumar Singh, y G. Giakos, Eds., Singapore: Springer, 2021, pp. 515-524. doi: 10.1007/978-981-16-5078-9_42.

[74] *5. Support Vector Machines*. Accedido: 13 de enero de 2025. [En línea]. Disponible en: https://learning.oreilly.com/library/view/hands-on-machine-learning/9781098125967/ch05.html

[75] A. F. Rasheed, M. Zarkoosh, S. F. Abbas, y S. S. Al-Azzawi, «TaskComplexity: A Dataset for Task Complexity Classification with In-Context Learning, FLAN-T5 and GPT-4o Benchmarks», 30 de septiembre de 2024, *arXiv*: arXiv:2409.20189. doi: 10.48550/arXiv.2409.20189.

[76] «GPT-4o mini: advancing cost-efficient intelligence». Accedido: 14 de enero de 2025. [En línea]. Disponible en: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[77] M. C. Suárez-Figueroa, I. Diab, E. Ruckhaus, y I. Cano, «First steps in the development of a support application for easy-to-read adaptation», *Univ Access Inf Soc*, vol. 23, n.º 1, pp. 365-377, mar. 2024, doi: 10.1007/s10209-022-00946-z.

[78] «Lectura fácil: Métodos de redacción y evaluación», Plena Inclusión Madrid. Accedido: 3 de febrero de 2025. [En línea]. Disponible en: https://plenainclusionmadrid.org/recursos/lectura-facil-metodos-redaccion-evaluacion/