Universidad Politécnica
de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

Grado en Matemáticas e Informática

Trabajo Fin de Grado

Lectura Fácil: Tiempos Verbales

Autor: Alejandro Peris Acedo

Tutora: María del Carmen Suárez de Figueroa Baonza

Cotutor: Isam Diab Lozano

Madrid, junio 2025

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

Trabajo Fin de Grado
Grado en Matemáticas e Informática
Título: Lectura Fácil: Tiempos Verbales
Junio 2025

Autor: Alejandro Peris Acedo

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

Cotutor:
Isam Diab Lozano
Departamento de Inteligencia Artificial (DIA)
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

La Lectura Fácil busca hacer los textos accesibles para personas con dificultades lectoras (discapacidades cognitivas, trastornos de aprendizaje o bajo nivel de alfabetización) transformando estructuras complejas, entre las que se incluyen formas verbales como los tiempos compuestos, los subjuntivos y los condicionales. Este Trabajo de Fin de Grado (TFG) desarrolla una solución tecnológica para identificar y adaptar automáticamente estas formas en español, alineándose con las directrices de Lectura Fácil [1] y sentando bases para sistemas de accesibilidad lingüística como FACILE [2].

En este TFG se ha propuesto diseñar un sistema para detectar formas verbales complejas mediante análisis morfosintáctico, transformar dichas estructuras en versiones más simples preservando el significado y evaluar su precisión y comprensibilidad, mediante un enfoque modular para facilitar su escalabilidad. Utilizando técnicas basadas en Procesamiento de Lenguaje Natural (PLN) el sistema se divide en dos módulos: uno identifica verbos complejos mediante etiquetas morfológicas y sintácticas, y otro los adapta con reglas predefinidas. Se complementa con una aplicación web que permite a los usuarios adaptar textos en tiempo real, probada con corpus reales. El módulo de identificación alcanzó un 77% de éxito, y el de adaptación transformó la mayoría de los verbos correctamente con una validez del 77% de los casos identificados.

Abstract

Easy-to-Read aims to make texts accessible for people with reading difficulties (cognitive disabilities, learning disorders, or low literacy levels) by transforming complex structures, including verb forms such as compound tenses, subjunctives, and conditionals. This project develops a technological solution to automatically identify and adapt these forms in Spanish, aligning with Easy-to-Read guidelines [1] and laying the foundation for linguistic accessibility systems like FACILE [2].

This TFG proposes designing a system to detect complex verb forms through morphosyntactic analysis, transform these structures into simpler versions while preserving meaning, and evaluate their accuracy and comprehensibility, using a modular approach to facilitate scalability. Employing Natural Language Processing (NLP) techniques, the system is divided into two modules: one identifies complex verbs using morphological and syntactic tags, and another adapts them with predefined rules. It is complemented by a web application that allows users to adapt texts in real-time, tested with real corpora. The identification module achieved a 77% success rate, and the adaptation module correctly transformed most verbs with a validity of 77% for identified cases.

Tabla de contenidos

Índice de Figuras

Índice de Tablas

1 Introducción

La Lectura Fácil es una metodología que promueve la creación de textos comprensibles para personas con dificultades de comprensión lectora, mediante pautas específicas relacionadas con el léxico, la gramática, la estructura textual y el diseño visual. Está dirigida a colectivos como personas con discapacidad intelectual, personas mayores, inmigrantes con dominio limitado del idioma o personas con bajo nivel educativo. Su origen se remonta a la publicación *Making Information Accessible to All* de la Comisión Europea en 1997 [6], y se ha consolidado en normativas como la UNE 153101:2018 EX [7], que garantizan el derecho universal a la información accesible.

Dentro de esta disciplina, uno de los principales retos lingüísticos es el uso de formas verbales complejas, como los tiempos verbales compuestos, el subjuntivo o el condicional, que dificultan la comprensión debido a su carga cognitiva, ya que exigen al lector procesar secuencias temporales complejas. Adaptar estas estructuras resulta esencial para adaptar textos a los principios de Lectura Fácil. En este contexto, el presente Trabajo de Fin de Grado propone un sistema basado en Procesamiento de Lenguaje Natural (PLN), utilizando la librería spaCy [3], para identificar y adaptar automáticamente formas verbales complejas en estructuras más simples, preservando el significado original.

Los objetivos principales de este trabajo son:

1. Identificar automáticamente tiempos verbales compuestos, subjuntivo y condicional en textos en español.
2. Adaptar dichas formas a tiempos verbales más simples y accesibles, manteniendo la coherencia semántica.
3. Evaluar la efectividad de las transformaciones en términos de comprensibilidad y fidelidad al contenido original.

Además de desarrollar un sistema funcional para la adaptación verbal, este trabajo tiene como objetivo secundario contribuir a un proyecto más amplio del departamento de Inteligencia Artificial, que busca integrar diversas directrices de Lectura Fácil para lograr la adaptación automática de textos. El sistema propuesto se diseñó con una arquitectura modular, compatible con enfoques similares desarrollados en iniciativas como el proyecto FACILE [2], lo que facilita su integración en soluciones más completas.

Este documento se estructura como sigue: el Capítulo 2 revisa el Estado de la Cuestión tanto de la Lectura Fácil como el Procesamiento de Lenguaje Natural; el Capítulo 3 describe la metodología empleada en el desarrollo del sistema y las pruebas que se han llevado a cabo; el Capítulo 4 presenta las conclusiones del proyecto; y el Capítulo 5 recoge las líneas de trabajo futuro.

2 Estado de la Cuestión

En este capítulo se presenta el contexto teórico y tecnológico sobre el que se apoya este TFG, dividido en tres secciones: la metodología de Lectura Fácil, las bases del Procesamiento de Lenguaje Natural y las herramientas empleadas en el desarrollo del sistema.

2.1 Metodología de Lectura Fácil

Las pautas de la metodología de la Lectura Fácil [1] son guías con el foco en personas con discapacidad cognitiva o con dificultades en la lectura la cual requiere de la creación y adaptación de textos siguiendo un criterio que ha estado evolucionando desde los años 90. Recientemente con el avance tecnológico se ha dado gran énfasis en la Lectura Fácil en muchos ámbitos seguido de la escritura de la norma UNE 153101:2018 EX [7] en dónde se recogen los criterios que tiene que seguir un texto para que cumpla con el objetivo planteado.

Estas directrices promueven la creación de textos claros y accesibles, priorizando un vocabulario sencillo, oraciones breves y estructuras gramaticales básicas. Dentro de estas directrices, se pueden encontrar ejemplos como:

- No abuses de las mayúsculas
- Evita el uso de términos abstractos, técnico o complejos
- Usa tiempos verbales simples y evita compuestos, condicionales y subjuntivos.

En concreto, este proyecto se centra en esta última recomendación sobre evitar tiempos verbales complejos.

La aplicación manual de estas directrices resulta costosa y pueden darse inconsistencias dependiendo de quien escriba o adapte el texto, especialmente en textos extensos o de alta complejidad. Por ello, se ha visto la necesidad de desarrollar soluciones tecnológicas que automaticen la adaptación de textos, integrando herramientas de inteligencia artificial para optimizar el proceso y garantizar resultados uniformes mientras se alivia la carga de trabajo para aquellos que adapten estos textos.

2.2 Procesamiento de Lenguaje Natural (PLN)

El Procesamiento de Lenguaje Natural es una rama de la Inteligencia Artificial en la cual se busca comprender el lenguaje humano tanto escrito como hablado. Para ello, se pueden usar múltiples técnicas, pero es necesario una base extensa

en lingüística. Razonablemente, este es un buen punto de partida para las bases de la automatización de la metodología de la Lectura Fácil ya que esta disciplina necesita una herramienta que permita entender el texto para poder adaptarlo dependiendo de cada pauta.

El PLN abarca un conjunto de técnicas destinadas a permitir que los sistemas computacionales comprendan y procesen el lenguaje humano. Este proyecto utiliza algunas de estas técnicas de PLN para cumplir con los requisitos de adaptación de textos a Lectura Fácil. A continuación, se describen dos técnicas fundamentales empleadas en este trabajo: la tokenización y el etiquetado.

2.2.1 Tokenización

Consiste en dividir el texto en unidades más pequeñas llamadas *tokens* que pueden ser desde palabras como "fácil", "leer" y "adaptación"; hasta signos de interrogación, exclamación, etc [8]. Este es el primer paso ya que es la base para identificar elementos gramaticales, como verbos y sus auxiliares. Un ejemplo del resultado de una tarea de tokenización sería la siguiente:

Ejemplo

"La adaptación de verbos es fundamental."

Tokenización:

```
[ ["La"], ["adaptación"], ["de"], ["verbos"], ["es"],
["fundamental"], ["."] ]
```

Es importante destacar que el punto final es su propio token ya que contiene información sobre el texto. Además, la tokenización no proporciona información suficiente para poder analizar la oración. Para ello, requerimos del uso del etiquetado.

2.2.2 Etiquetado Gramatical (*Part-of-Speech*)

Tras la tokenización, pasamos a la siguiente técnica, el etiquetado gramatical (*Part-of-Speech* o POS), que consiste en asignar a cada token una categoría gramatical o POS (verbo, sustantivo, adjetivo, etc.) y sus respectivas propiedades morfológicas, como tiempo, modo, persona o número en el caso de los verbos [9]. De esta forma, podemos obtener una mayor cantidad de información sobre la frase y, como hemos mencionado, podemos distinguir entre las formas verbales, como el pretérito anterior o el pretérito perfecto compuesto.

Es necesario recalcar la existencia de varios tipos de etiquetas. La más simple indica las categorías gramaticales generales o *Universal Part-of-Speech* (UPOS) (ADJ para adjetivos, VERB para verbos, etc.) [10], pero se necesita información adicional para obtener esa distinción entre formas verbales. Debido a esto, se utiliza un etiquetado morfológico o *Universal Features* [11] proporcionando así

pequeños matices como el género para sustantivos o el modo para verbos. Una última etiqueta que será utilizada es el lema que contiene la forma canónica o base de una palabra mediante un proceso que se conoce como lematización.

Siguiendo con el ejemplo anterior podemos obtener sus etiquetas en la siguiente tabla:

Tabla 1 Ejemplo de etiquetado y lematización de una oración

Token	***Lemma***	***POS***[1]	***Morph***

La	El	DET	`Definite=Def`
			`Gender=Fem`
			`Number=Sing`
			`PronType=Art`

Adaptación	Adaptación	NOUN	`Gender=Fem` `Number=Sing`
De	De	ADP	
Verbos	Verbo	NOUN	`Gender=Masc` `Number=Plur`

Es	Ser	AUX	`Mood=Ind`
			`Number=Sing`
			`Person=3`
			`Tense=Pres`
			`VerbForm=Fin`

Fundamental	Fundamental	ADJ	`Number=Sing`
.	.	PUNCT	`PunctType=Peri`

2.3 Herramientas

El desarrollo del sistema para identificar y adaptar tiempos verbales complejos en textos en español se apoya en un conjunto de herramientas tecnológicas que permiten procesar textos de manera eficiente y cumplir con las directrices de Lectura Fácil. Estas herramientas son Python, spaCy, FastAPI y Vite, seleccionadas por su capacidad para abordar el análisis lingüístico y la interacción con usuarios. Entre ellas, spaCy destaca como la herramienta principal para el procesamiento del lenguaje natural, mientras que FastAPI y

[1] Se ha utilizado las abreviaturas inglesas usadas en spaCy.
POS son las etiquetas gramaticales.
Morph son las etiquetas morfológicas.

Vite cumplen funciones complementarias en la creación de una interfaz accesible.

2.3.1 Python

Python [12] es un lenguaje de programación ampliamente utilizado en el ámbito del PLN, sirve como la base del sistema debido a su simplicidad y su extenso ecosistema de librerías. En este proyecto, Python coordina todas las tareas, desde la carga de textos hasta la integración de los resultados adaptados, permitiendo una implementación flexible y escalable que automatiza la adaptación de textos.

2.3.2 spaCy

spaCy [3], una biblioteca de código abierto para el PLN. Es la herramienta central para analizar y transformar textos. Su modelo para el español, basado en arquitecturas transformadoras (es_dep_news_trf), ofrece alta precisión en tareas como la tokenización, el etiquetado morfosintáctico y la lematización. Se han hecho pruebas para analizar el funcionamiento de los distintos modelos que ofrece spaCy, pero se ha tomado la decisión de utilizar el modelo transformador ya que ofrece una precisión insuperable con una penalización asumible en términos de eficiencia.

En este trabajo, spaCy se utiliza para segmentar textos en tokens, identificar verbos y sus auxiliares, y determinar sus propiedades morfológicas, tal y como se ha expuesto en las secciones anteriores sobre el PLN. La capacidad de spaCy para personalizar reglas de análisis permite adaptar el procesamiento a tiempos verbales complejos, como el subjuntivo o el condicional, asegurando que las transformaciones sean precisas y alineadas con las directrices de Lectura Fácil. Su eficiencia y robustez han sido fundamentales para reducir la carga de trabajo manual y garantizar resultados uniformes en la adaptación de textos.

2.3.3 FastAPI

FastAPI [4] es un *framework*[2] de Python para crear APIs[3] web, se emplea para exponer las funcionalidades del sistema como un servicio accesible. En este proyecto, FastAPI recibe textos enviados por los usuarios, los procesa utilizando spaCy y devuelve los resultados adaptados, actuando como un enlace entre el análisis lingüístico y la interfaz de usuario. Su rol es secundario, pero facilita la integración del sistema en un entorno web.

---

[2] Un entorno de trabajo o *framework* es una estructura o conjunto de herramientas y componentes que facilitan el desarrollo de aplicaciones o páginas web

[3] Una API (*Application Programming Interface* en inglés) o Interfaz de Programación de Aplicaciones es un conjunto de reglas y protocolos que permiten a diferentes aplicaciones comunicarse entre sí.

2.3.4 Vite

Vite [5] es una herramienta de desarrollo frontend que se utiliza para crear una interfaz web sencilla que permite a los usuarios interactuar con el sistema. Basada en módulos ES, Vite ofrece un entorno optimizado para construir aplicaciones web rápidas y ligeras.

En este proyecto, Vite se emplea para desarrollar una interfaz que muestra los textos originales y sus versiones adaptadas, generadas por el sistema. Al igual que FastAPI, su rol es complementario, proporcionando un medio visual para que los usuarios, como editores o adaptadores de textos, puedan utilizar la herramienta de manera práctica, en línea con los principios de accesibilidad de la Lectura Fácil.

2.3.5 Integración de herramientas

La combinación de Python [12], spaCy [3], FastAPI [4] y Vite [5] forma un sistema cohesionado que cubre todas las etapas del proyecto. Python actúa como el núcleo que unifica las herramientas, spaCy proporciona el análisis lingüístico avanzado para identificar y transformar tiempos verbales, mientras que FastAPI y Vite ofrecen soporte para la interacción con los usuarios a través de una API y una interfaz web, respectivamente. Este enfoque integrado garantiza que el sistema sea eficiente, escalable y capaz de producir textos adaptados que cumplen con los criterios de la metodología de la Lectura Fácil, aliviando la carga de trabajo manual y asegurando resultados consistentes.

3 Desarrollo

En este capítulo, se profundizará en la fase de desarrollo mencionando el diseño, la implementación y las pruebas del sistema en su totalidad. En cada sección se expondrá los desafíos encontrados y las soluciones propuestas en la realización de este trabajo.

3.1 Diseño del proyecto

Como ya hemos mencionado, nos centraremos en la recomendación de evitar tiempos verbales complejos. Más concretamente buscamos evadir el uso de formas verbales compuestas, el modo subjuntivo y la forma condicional. El problema se centra en dos desafíos principales.

En primer lugar, es necesario detectar con precisión tiempos verbales complejos. En este proyecto, nos centraremos exclusivamente en las formas verbales siguientes:

Modo Indicativo

- Pretérito perfecto compuesto
- Pretérito anterior
- Pretérito pluscuamperfecto
- Futuro compuesto
- Condicional simple
- Condicional compuesto

Modo Subjuntivo

- Pretérito perfecto compuesto
- Pretérito pluscuamperfecto
- Futuro compuesto

Estos verbos, que en su mayoría contienen verbos auxiliares y participios, requieren un análisis lingüístico detallado para identificar sus propiedades morfológicas, como tiempo, modo, persona y número. Por otro lado, las perífrasis verbales han sido tratadas en trabajos anteriores [13] [14] y, por lo tanto, escapan del alcance de este proyecto. Del mismo modo, solo utilizaremos oraciones en voz activa ya que en este TFG nos centramos en la pauta de Lectura Fácil sobre los tiempos verbales complejos y la voz pasiva cuenta con su propia pauta distinta. Además, la complejidad añadida no puede ser asumida por este Trabajo de Fin de Grado. Veamos a continuación un ejemplo de identificación:

Ejemplo

"Si hubieras llegado antes, te habrías divertido"

En la oración anterior, se debe reconocer “hubieras llegado” como subjuntivo pluscuamperfecto y “habrías divertido” como condicional compuesto.

En segundo lugar, una vez identificados, estos tiempos verbales deben transformarse en formas más simples, como el pretérito perfecto simple o el presente de indicativo. En este TFG, se busca preservar el significado original del texto en la medida de lo posible, pero, debido a la propia naturaleza de la lengua española, en casos particulares la sustitución de verbos puede diluir la intención original del texto.

Este proceso implica definir reglas de transformación que sean consistentes y aplicables a diferentes contextos.

Ejemplo

Original: “*Hoy he terminado el informe*”

Adaptado: “*Hoy terminé el informe*”

En este ejemplo podemos ver que transformamos un tiempo verbal del pretérito perfecto compuesto al pretérito perfecto simple siguiendo las guías de la Lectura Fácil.

El idioma español consta con múltiples excepciones e irregularidades lo cual presenta un grave desafío a la hora de realizar esta tarea. Un gran número de verbos son irregulares y puede llevar a ambigüedades contextuales. Por ejemplo, analicemos la siguiente oración:

Ejemplo

“Pero jamás las habría imaginado juntas, si ella…”

Al no tener contexto, el verbo “*habría imaginado*” podría ser primera persona del singular o tercera persona del singular. La interpretación del primer caso es la más común debido al poco contexto y a la naturaleza del verbo “imaginar”, puesto que, para poder decir esta frase en tercera persona, debes de conocer en detalle a la otra persona para poder saber lo que piensa.

Esta ambigüedad obliga el uso de herramientas de PLN capaces de analizar la estructura sintáctica y semántica de las oraciones. Además, como objetivo adicional, buscamos un sistema que sea eficiente para procesar textos extensos y garantizar resultados uniformes.

Para la obtención de textos de prueba que contengan los tiempos verbales complejos mencionados, se ha utilizado el Corpus del Español del Siglo XXI (CORPES), gestionado y recopilado por la Real Academia Española [15]. Este corpus, que recoge muestras de la lengua española del siglo XXI, permite filtrar

textos por tiempos verbales específicos, asegurando que cada texto incluya al menos un verbo en la forma deseada, como el pretérito pluscuamperfecto o el condicional compuesto. Esta funcionalidad ha sido de gran ayuda para validar la identificación y transformación de tiempos verbales en contextos reales, proporcionando una base diversa y representativa para el desarrollo y evaluación del sistema.

Para abordar este problema, se propone el desarrollo de un sistema basado en las herramientas de PLN, con un enfoque particular en spaCy. Esta biblioteca permite realizar tareas como la tokenización, el etiquetado morfosintáctico y la lematización, que nos darán la información necesaria para la identificación y adaptación que hemos expuesto previamente.

3.2 Implementación

3.2.1 Identificación de verbos complejos

Como se ha mencionado anteriormente, el primer paso de este proyecto es la identificación de tiempos verbales complejos siguiendo el diseño propuesto. Este proceso se centra en reconocer las formas compuestas, el modo subjuntivo y el condicional, listadas en la sección 3.1. Para ello, usaremos el modelo transformador de spaCy.

En primer lugar, hacemos uso de la tokenización de spaCy para obtener el análisis morfosintáctico del texto. Esto dividirá todas las palabras con sus propias etiquetas que utilizaremos en nuestro algoritmo para identificar los verbos. Para buscar estos verbos, recorremos la lista de tokens creada por spaCy en orden hasta detectar una etiqueta de verbo o de auxiliar de verbo. Empezamos buscando los casos que tengan verbos compuestos ya que, en general, tienen una estructura similar a esta:

Ejemplo

"El gerente había pedido un trabajo."
AUX VERB

Por lo tanto, el primer token que detectemos es un auxiliar. Si a este auxiliar le precede un verbo, asumimos que es un verbo compuesto, aunque existe una forma más elegante de identificar un auxiliar de verbo. Entre las etiquetas que proporciona spaCy existe una etiqueta de dependencia que nos dice de quien depende cada palabra. En este caso, "había" depende de "pedido" que define la acción tomada, lo cual será importante más adelante en la adaptación.

En el caso de que no sea compuesto, primero identificamos un token con la etiqueta de verbo sin auxiliares y después, utilizamos las etiquetas morfológicas para identificar la forma verbal (un ejemplo de estas etiquetas se puede encontrar en la sección 2.2.2).

Una vez identificados los verbos, buscaremos clasificarlos para facilitar el trabajo de la adaptación. Volvemos a hacer uso de las etiquetas de spaCy para identificar el tiempo y el modo de cada verbo obteniendo así una explicación de la identificación del verbo como complejo.

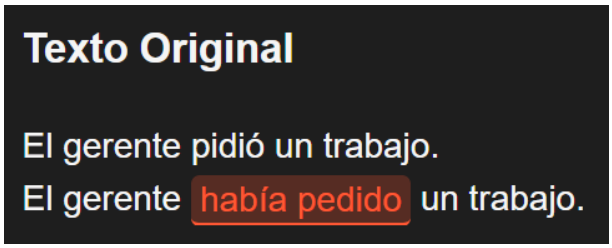

Figura 1 Identificación de un verbo compuesto por el sistema

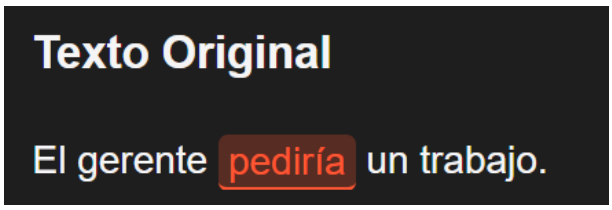

Figura 2 Identificación de un verbo condicional por el sistema

En la figura 1 y 2 podemos ver el análisis consecutivo de tres oraciones mediante el sistema desarrollado. En la primera figura vemos resaltado el verbo "había pedido" ya que se trata de la forma verbal del pretérito pluscuamperfecto, mientras que el verbo simple "pidió" se mantiene sin resaltar. En la siguiente imagen, se puede observar también el resalto de "pediría" por ser la forma simple del condicional.

Este proceso se hace una única vez para cada texto ya que, al recorrer toda la lista de tokens, permitimos la identificación simultánea de múltiples verbos problemáticos. Es decir, el sistema de identificación está diseñado para poder analizar cualquier texto y poder identificar todos los verbos complejos. Debido a esto la complejidad supera O(n), aunque por la propia naturaleza de los textos españoles, los verbos representan un bajo porcentaje de las palabras en un texto cualquiera. Por ello, la ejecución de este algoritmo es rápida, aunque no sea eficiente.

Por último, la salida de este sistema es una lista de problemas encontrados, donde cada problema es un verbo complejo que deberá ser adaptado. Cada problema incluye una explicación con la forma verbal detectada y unos índices de inicio y final del problema para localizar cada uno de los verbos.

3.2.2 Adaptación del texto

Una vez identificados los verbos complejos, el siguiente paso es adaptarlos a formas más simples para cumplir con las directrices de la Lectura Fácil. Este proceso transforma tiempos verbales compuestos, subjuntivos y condicionales en formas más accesibles preservando el significado original del texto.

Este sistema requiere de entrada la salida del módulo de identificación junto con el propio texto original. Para empezar, es necesario analizar la salida del módulo anterior para obtener la forma verbal, que se encuentra en la explicación, y los índices de inicio y fin del problema.

Con esta información, hacemos uso una vez más de spaCy para obtener las etiquetas de los tokens dentro de los índices problemáticos. Dentro de este rango, procesamos el texto en busca de un auxiliar y un verbo o solamente un verbo. Esto tiene como finalidad buscar el verbo principal que contiene la acción de la oración y, si existe, un verbo auxiliar. Esta doble comprobación también nos permite encontrar problemas que se originen en el módulo anterior, pero no hayan sido detectados satisfactoriamente.

Al procesar el texto problemático, obtenemos dos posibles casos: el verbo es compuesto y por ello la información está dividida entre el verbo auxiliar y el principal o el verbo es simple y contiene toda la información necesaria.

El caso más simple es aquel con solo un verbo ya que toda la información recae en ese mismo verbo, tanto el lexema o infinitivo que tenemos que usar como la persona y el número. Esto se puede observar en la siguiente tabla:

Tabla 2 Ejemplo de verbo simple con sus etiquetas

Token	**Lema**	**POS**	**Morph**

Pediría	Pedir	VERB	`Mood=Cnd`
			`Number=Sing`
			`Person=3`
			`VerbForm=Fin`

Por otro lado, si el verbo es compuesto, tenemos que buscar la persona y el número dentro de las etiquetas del verbo auxiliar.

Tabla 3 Ejemplo de verbo compuesto con sus etiquetas

Token	**Lema**	**POS**	**Morph**
Había	Haber	AUX	Mood=Ind Number=Sing Person=3 Tense=Imp VerbForm=Fin
Pedido	Pedir	VERB	Gender=Masc Number=Sing Tense=Past VerbForm=Part

Se puede apreciar en la tabla anterior que la persona y número (`Person=3` y `Number=Sing`) están en el verbo auxiliar "Había", pero el lexema que tenemos que usar para la transformación es "Pedir" que viene del verbo principal.

En este punto hemos obtenido toda la información necesaria para la transformación:

- Forma verbal actual, obtenido en la identificación.
- Persona
- Número
- Infinitivo

Para poder aplicar la transformación de verbos hemos hecho uso de un algoritmo de reglas. Es decir, para cada forma verbal compleja le corresponde una forma verbal simple que sigue las indicaciones de la Lectura Fácil.

Tabla 4 Transformaciones aplicadas para cada forma verbal compleja del indicativo

Modo	**Tiempo compuesto**	**Tiempo simple**	**Frase original**	**Frase adaptada**
Indicativo	Pretérito perfecto compuesto	Pretérito perfecto simple	Hoy he terminado el informe.	Hoy terminé el informe.
	Pretérito anterior	Pretérito perfecto simple	Cuando hube llegado, ya se habían ido.	Cuando llegué, ya se habían ido.
	Pretérito pluscuamperfecto	Pretérito perfecto simple	Había salido cuando llegaste.	Salí cuando llegaste.
	Futuro compuesto	Futuro simple	El lunes habré terminado el informe.	El lunes termino el informe.
	Condicional compuesto	Pretérito imperfecto	Si hubieras venido antes, te habrías divertido.	Si venías antes, te divertías.
	Condicional simple	Pretérito imperfecto	En tu lugar, yo hablaría con el profesor	En tu lugar, yo hablaba con el profesor

Tabla 5 Transformaciones aplicadas para cada forma verbal compleja del subjuntivo

Modo	**Tiempo compuesto**	**Tiempo simple**	**Frase original**	**Frase adaptada**
Subjuntivo	Pretérito perfecto compuesto	Pretérito perfecto simple de indicativo	Dudo que Juan haya terminado a tiempo.	Dudo que Juan termine a tiempo.
	Pretérito pluscuamperfecto	Pretérito imperfecto de indicativo	Si hubiera sabido la respuesta, la habría dicho.	Si sabía la respuesta, la decía.
	Futuro compuesto	Presente indicativo	Cuando hubiere terminado el informe, lo enviaré.	Cuando acabe el informe, lo envío.

Siguiendo las reglas de las tablas 4 y 5, podemos aplicar toda la información obtenida para conseguir la conjugación del nuevo verbo que siga las pautas de la Lectura Fácil.

La conjugación está diseñada mediante un sistema independiente en el cuál hacemos uso de una API no oficial de la RAE [16] para obtener la definición y todas las conjugaciones de un verbo. A continuación, podemos ver un ejemplo de conjugación haciendo uso de esta API:

Presente

Persona	Singular	Plural
1ª (yo/nosotros)	pido	pedimos
2ª (tú/vosotros)	pides / pedís	pedís
2ª formal (usted/ustedes)	pide	piden
3ª (él,ella/ellos,ellas)	pide	piden

Figura 3 Ejemplo de la conjugación del presente de pedir en rae-api

Obtenemos el verbo adaptado con la información que hemos obtenido y el sistema de conjugaciones descrito previamente que será reemplazado por el verbo complejo en el texto original, obteniendo así un nevo texto adaptado. Aun así, este proceso genera un nuevo problema por la naturaleza de las cadenas de caracteres en Python.

El módulo de adaptación descrito hasta ahora solo funciona para un verbo complejo y al transformar el texto, los índices de inicio y final de los verbos posteriores se verán afectados por esta transformación ya que es muy probable que la longitud del verbo remplazado sea mayor que el verbo adaptado. Por ello, es necesario dos medidas para el correcto funcionamiento.

Primero, actualizamos el índice final del verbo adaptado para que se ajuste a su nuevo tamaño. Segundo, para cada índice de inicio y final de cada verbo posterior a él, restamos la diferencia entre el índice final original del verbo adaptado y el nuevo índice final. En otras palabras, desplazamos todos los índices a la derecha o izquierda dependiendo de cómo de largo o corto sea el nuevo verbo. Para mayor claridad, se presenta un ejemplo con todos los pasos que sigue nuestro sistema:

Ejemplo

Texto Original

Me *ha gustado* mucho, pero me *debería* volver a casa. *He estado* mucho tiempo fuera.

Tabla 6 Problemas detectados en el texto original de ejemplo

	Verbo complejo 1	**Verbo complejo 2**	**Verbo complejo 3**
Verbo	"ha gustado"	"debería"	"He estado"
Explicación	Pretérito Perfecto Compuesto	Condicional Simple	Pretérito Perfecto Compuesto
Posición Inicio	3	29	52
Posición Final	13	36	61

Primera iteración de la adaptación

Me *gustó* mucho, pero me *debería* volver a casa. *He estado* mucho tiempo fuera.

Tabla 7 Actualización de índices tras la primera iteración del módulo de adaptación

	Verbo complejo 1	**Verbo complejo 2**	**Verbo complejo 3**
Verbo	"gustó"	"debería"	"He estado"
Explicación	Pretérito Perfecto Simple	Condicional Simple	Pretérito Perfecto Compuesto
Posición Inicio	3	24	52
Posición Final	8	31	61

Segunda iteración de la adaptación

Me *gustó* mucho, pero me *debo* volver a casa. *He estado* mucho tiempo fuera.

Tabla 8 Actualización de índices tras la segunda iteración del módulo de adaptación

	Verbo complejo 1	**Verbo complejo 2**	**Verbo complejo 3**
Verbo	"gustó"	"debo"	"He estado"
Explicación	Pretérito Perfecto Simple	Presente	Pretérito Perfecto Compuesto
Posición Inicio	3	24	44
Posición Final	8	28	53

En la primera iteración, se adapta el primer verbo “ha gustado” a “gustó”, la diferencia de tamaño entre el primer y segundo verbo es de 5 caracteres por lo que restamos esta misma cantidad a la posición de inicio (29) y final (36) del verbo “debería” obteniendo 24 y 31 respectivamente. Para mayor eficiencia, se puede ver solo se actualiza la posición del tercer verbo cuando es el siguiente verbo en ser adaptado. Esto se hace sumando las desviaciones obtenidas hasta ese momento, que en este caso sería 8 caracteres; 5 del primer verbo y 3 del segundo.

3.2.3 Aplicación

Para la realización de este proyecto y poder demostrar sus resultados, se ha diseñado una aplicación web sencilla que permite interactuar con ambos módulos diseñados en las secciones previas. Esta aplicación ha sido desarrollada en Vite para la interfaz gráfica y FastAPI para unir ambos módulos y dar respuesta a la interfaz y su función es integrar el trabajo realizado hasta ahora.

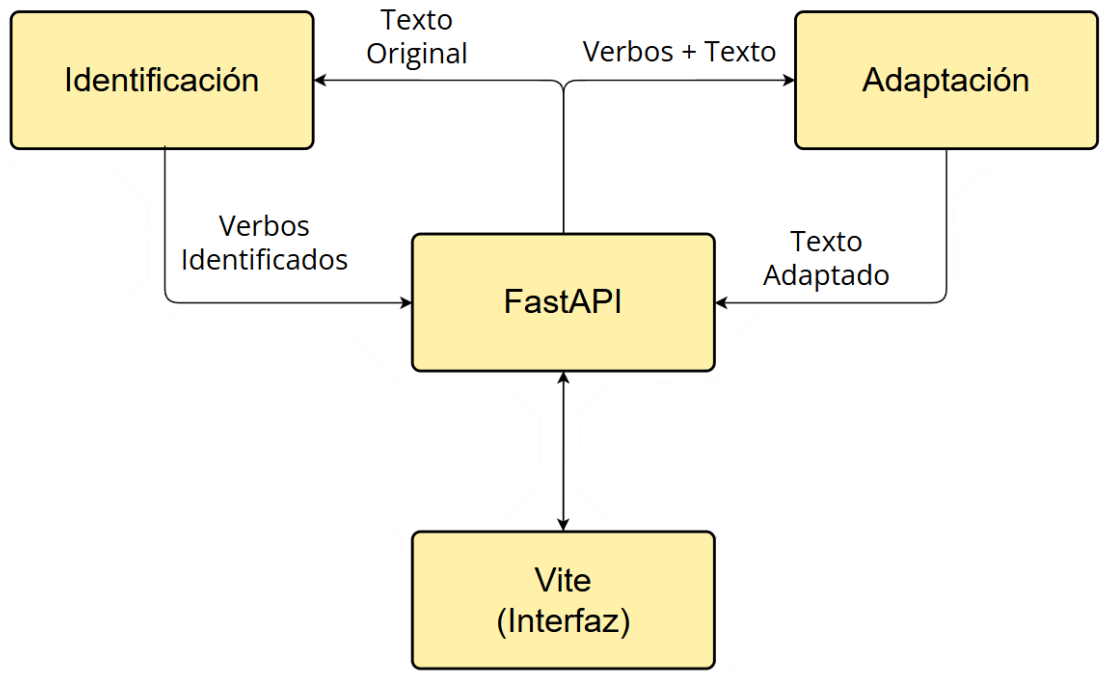

Figura 4 Diseño en diagrama de flujo de la aplicación de este TFG[4]

Como podemos ver en la figura 4, FastAPI hace de módulo central donde se maneja la mayoría de la información. Se encarga de transmitir la información de un módulo al siguiente y de transformar dicha información para que la interfaz sea capaz de entenderla y mostrarla. Este backend cuenta con tres funciones principales. Dos corresponden con cada uno de los módulos de identificación y adaptación mientras que la tercera función es un buscador local del corpus de la RAE mencionado previamente que usaremos para las pruebas del sistema.

[4] Este diagrama de flujo ha sido realizado mediante una herramienta online gratuita llamada: *Visual Paradigm*.
https://online.visual-paradigm.com/es/diagrams/solutions/free-flowchart-maker/

La función principal de la aplicación es ofrecer una forma rápida de visualizar los resultados. Es por ello, que usamos una interfaz limpia que permita ejecutar pruebas del código de forma eficiente. Dentro de esta interfaz podemos apreciar un cuadro de texto para poder analizar cualquier oración o texto, siendo este nuestro principal método de interacción con el sistema. Tras escribir el texto, pulsamos el botón de "Analizar" que hará la primera llamada al backend de FastAPI. Este transmite la información al módulo de identificación y nos devuelve el análisis correspondiente a este texto. Todos los verbos complejos detectados serán resaltados en rojo con una explicación al poner el cursor encima cómo se puede apreciar en la figura 5.

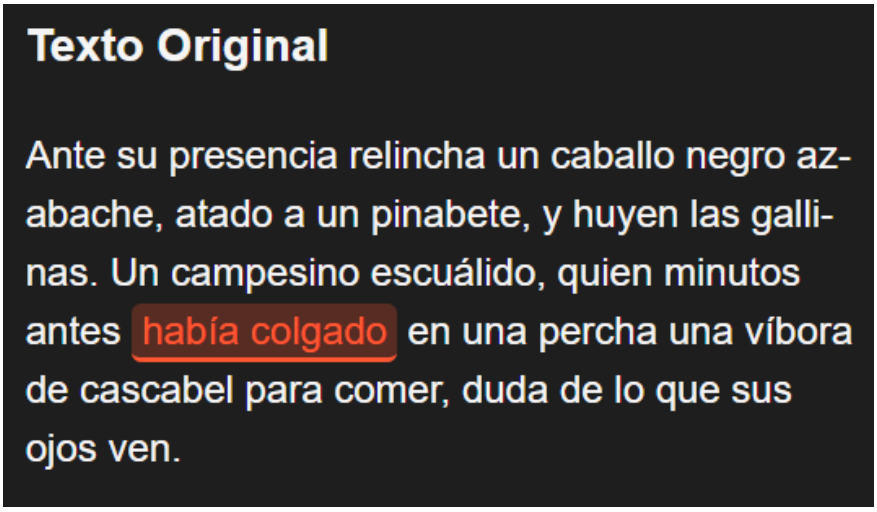

Figura 5 Ejemplo del análisis dentro de la aplicación

Tras esto, se permite pulsar un nuevo botón que genera otra llamada al backend para el segundo módulo de adaptación. De nuevo, FastAPI transmite la información, en este caso el texto y sus problemas detectados, a este sistema que devuelve un nuevo texto adaptado con los verbos transformados cómo se puede observar en la figura 6.

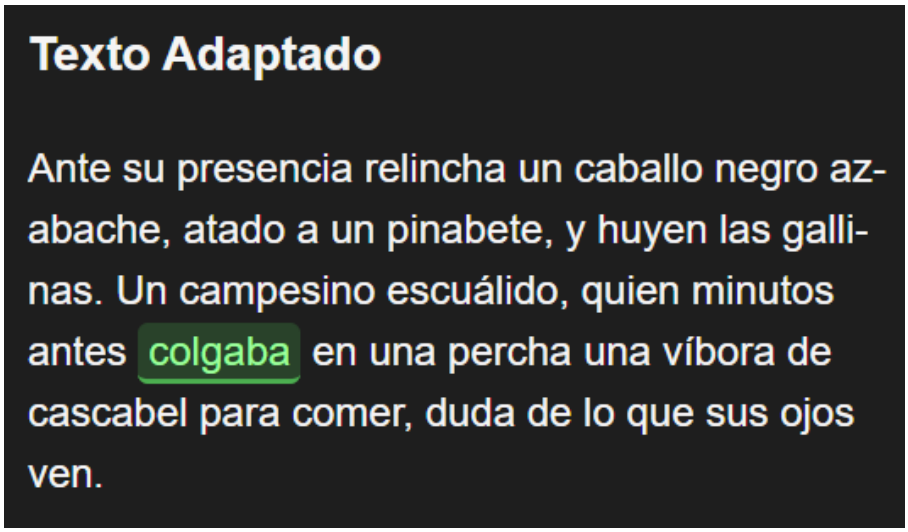

Figura 6 Ejemplo de la adaptación dentro de la aplicación

Para facilitar las pruebas, la aplicación cuenta con un sistema de consulta de textos con tiempos verbales específicos clasificado por la RAE. Esto permite usar estos textos en nuestro sistema para probar la efectividad del conjunto de la aplicación. Para acceder a este corpus se ha descargado de forma local una pequeña segmentación de ejemplos de cada tiempo verbal analizado. Cada tiempo verbal cuenta con entre 700 y 900 ejemplos distintos. El diseño de esta búsqueda se puede apreciar en la siguiente figura:

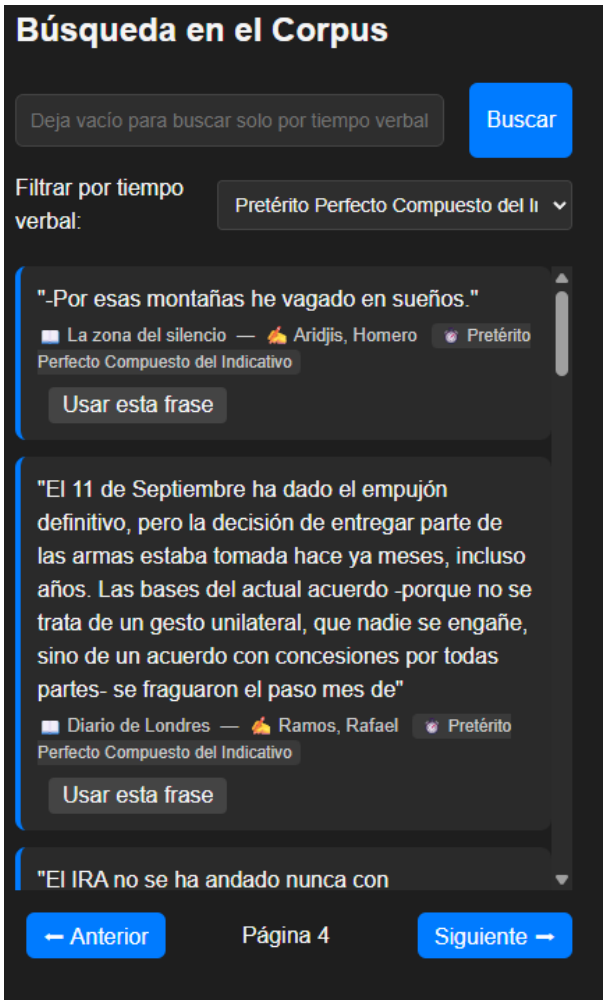

Figura 7 Búsqueda del Pretérito Perfecto Compuesto en la aplicación.

3.3 Pruebas

3.3.1 Corpus del Español del Siglo XXI (CORPES)

Como se ha expuesto con anterioridad, el Corpus del Español del Siglo XXI [15] ha sido fundamental para las pruebas requeridas en este trabajo. Este corpus, además de ser implementado en la aplicación como una ayuda para las pruebas rápidas, ha sido de gran ayuda para desarrollar las estadísticas y métricas de este Trabajo de Fin de Grado. Aun así, este Corpus no está exento de problemas que a continuación exponemos.

Dentro del CORPES, puedes filtrar todos los documentos disponibles por su categoría gramatical. En nuestro caso, ha sido utilizado para buscar verbos con tiempos verbales específicos, como el pretérito anterior o el pretérito pluscuamperfecto del subjuntivo. Esta consulta nos devuelve una serie de textos que podemos ver directamente en su página web o pueden ser descargados localmente. Cuando se muestran en la página web podemos ver que se muestra el texto que coincide con la búsqueda junto con la palabra específica que coincide con el filtro que se ha aplicado para esa búsqueda, pero al descargarse localmente un archivo TSV, perdemos esta palabra que coincide con la búsqueda. Por lo tanto, se obtiene un fichero con textos y sus metadatos, pero sin la palabra específica que permitió filtrar el propio texto.

En este trabajo, hemos descargado un segmento de esta base de datos cuyo listado es el siguiente:

- Condicional compuesto
- Condicional simple
- Futuro compuesto del indicativo
- Presente del indicativo
- Pretérito anterior del indicativo
- Pretérito imperfecto del subjuntivo terminados en -ra
- Pretérito imperfecto del subjuntivo terminados en -se
- Pretérito perfecto compuesto del indicativo
- Pretérito perfecto compuesto del subjuntivo
- Pretérito pluscuamperfecto del indicativo
- Pretérito pluscuamperfecto del subjuntivo terminado en -ra
- Pretérito pluscuamperfecto del subjuntivo terminado en -se

Algunos de estos datos no deberían contener ningún problema, pero al filtrar por textos que contengan, por ejemplo, el presente del indicativo este texto solo me garantiza que existe al menos un verbo en presente del indicativo dentro del texto. Por ello, el texto puede contener otros verbos de distintos tiempos verbales.

Ejemplo

A continuación, se puede ver un extracto obtenido haciendo uso del filtro "*Presente del Indicativo*"

Año: 2001.
Criterio: Fecha de escritura.
Medio: Escrito.
Bloque: No ficción.
Soporte: Web.
Tema: Actualidad, ocio y vida cotidiana.
País: España.
Tipología: Blog

Ramos, Rafael: «Castillos, caravanas y estrellas Michelín». Diario de Londres. www.blogs.lavanguardia.com: blogs.lavanguardia.com, 2001-06-25.

"…el 'Stagg Inn', antigua posada para las diligencias, que ha hecho *historia como el primero que* recibe *una estrella Michelín."*

El fragmento anterior perteneciente a Rafael Ramos ha sido filtrado mediante el presente del indicativo por el verbo "recibe", pero contiene además un verbo "ha hecho" cuyo tiempo verbal es pretérito perfecto compuesto y, por lo tanto, este fragmento sería detectado como problemático.

Podemos sacar dos conclusiones de este problema. La primera conclusión que obtenemos es que los textos de este Corpus pueden contener verbos adicionales y segundo, estos verbos pueden ser de tiempos verbales distintos al filtro que se aplicó en un principio. Como veremos más adelante, esto dificulta las pruebas.

3.3.2 Elección del modelo

Uno de los primeros pasos en el desarrollo de este proyecto fue determinar qué modelo de spaCy sería el más adecuado para las tareas de procesamiento del lenguaje natural requeridas, específicamente para la detección de verbos. Según la documentación oficial de spaCy, el modelo pequeño está optimizado para una mayor velocidad, lo que lo hace ideal para aplicaciones con recursos limitados o necesidades de procesamiento rápido, mientras que el modelo de transformadores está diseñado para ofrecer una mayor precisión, aunque a costa de un mayor consumo computacional. Para tomar una decisión informada, se llevaron a cabo pruebas comparativas entre ambos modelos utilizando un extracto del CORPES de 50 textos.

El objetivo principal de estas pruebas fue evaluar la capacidad de cada modelo para detectar correctamente los verbos en los diálogos, ya que la identificación

precisa de los verbos es fundamental para la posterior adaptación de estos en formato de Lectura Fácil. Las pruebas se diseñaron para analizar tanto oraciones con estructuras sintácticas simples como oraciones más complejas que incluían verbos con posibles dobles significados o construcciones verbales compuestas, donde los modelos podrían mostrar diferencias significativas en su rendimiento.

Los resultados muestran que el modelo de transformadores logró detectar correctamente el 100% de los verbos en las oraciones analizadas, mientras que el modelo pequeño alcanzó una precisión del 84% o 42 oraciones analizadas correctamente frente a las 50 oraciones identificadas correctamente por el modelo transformador. Aunque el modelo pequeño mostró una velocidad de procesamiento significativamente mayor (aproximadamente un 40% más rápido en promedio), la diferencia en precisión fue un factor determinante, especialmente en oraciones con estructuras más complejas.

Ejemplo

"Vuelvo a la vida con la muerte al hombro, abominando cuanto he escrito."

En esta oración, el modelo de transformadores identificó correctamente todos los verbos: «vuelvo» (presente de indicativo, primera persona singular), «he escrito» (pretérito perfecto compuesto, primera persona singular) y «abominando» (gerundio). Además, asignó las etiquetas morfosintácticas correctas, incluyendo tiempo, modo y persona, lo que es fundamental para determinar cómo transformar estas formas verbales en un formato más simple. En cambio, el modelo pequeño no detectó el verbo «vuelvo», debido a su limitada capacidad para resolver dependencias sintácticas complejas en oraciones con múltiples verbos o estructuras coordinadas. Este fallo afectó la asignación de etiquetas, lo que podría derivar en transformaciones incorrectas de los verbos en el formato de Lectura Fácil.

La precisión en la detección y etiquetado de verbos complejos es un requisito fundamental para este proyecto, ya que las transformaciones de verbos dependen directamente de su correcta identificación, tanto del verbo como de sus etiquetas morfosintácticas. Aunque el modelo pequeño ofrece ventajas en términos de velocidad, la diferencia del 16% en precisión resultó inaceptable, especialmente en oraciones con verbos de doble significado o estructuras compuestas. La diferencia en velocidad de procesamiento fue considerada asumible, ya que las pruebas se realizaron en un entorno con recursos computacionales suficientes para soportar el modelo de transformadores. Por lo tanto, se decidió utilizar el modelo de transformadores para el resto del proyecto, priorizando la precisión y la calidad del etiquetado sobre la velocidad de procesamiento.

3.3.3 Pruebas del módulo de identificación

El módulo de identificación de verbos complejos ha sido evaluado mediante pruebas haciendo uso del CORPES. El objetivo principal de estas pruebas es comprobar la efectividad del sistema para detectar y clasificar las formas verbales complejas frente a formas simples que sirvieron como nuestro control. Estas categorías de control, en teoría, no deberían ser detectadas como problemáticas, ya que representan formas verbales más accesibles en el contexto de la Lectura Fácil. Sin embargo, como se mencionó anteriormente, el corpus de la RAE presentó desafíos debido a la presencia de verbos adicionales en los fragmentos filtrados, lo que afectó la evaluación.

Para las pruebas, se analizaron 210 fragmentos de texto extraídos del corpus, de los cuales 100 correspondieron a la categoría de control "Presente del Indicativo" y 110 a diversas categorías de tiempos verbales complejos, incluyendo pretérito perfecto compuesto, pluscuamperfecto, futuro compuesto, condicional simple y compuesto, y varias formas del subjuntivo. Cada fragmento fue procesado por el módulo de identificación que devolvió una lista de problemas detectados, donde cada problema incluía el verbo identificado, su forma verbal y una explicación de por qué se consideró complejo, tal y como está descrito en la sección 3.2.1.

Para mitigar los efectos que tiene el desafío del corpus, se ha tomado arbitrariamente el filtro de Presente del Indicativo y se ha limpiado manualmente los 100 fragmentos utilizados en las pruebas de este módulo. Por ello, existe la posibilidad que varios de los “False Positives” sean verdaderamente fragmentos de texto con verbos complejos que no han sido propiamente filtrados. Esto puede afectar la precisión de las pruebas y dar un peor resultado debido a una base de datos con fallos.

Los resultados generales muestran que el módulo de identificación detectó problemas en 119 de los 210 fragmentos analizados, lo que representa una tasa de detección del 56,67% similar a la división entre extractos de control y problemáticos. La precisión global fue del 76,47%. A continuación, desglosaremos cada tipo de fallo o acierto:

- Verdaderos Positivos: El sistema identificó correctamente 91 casos de verbos complejos. Por ejemplo, tomemos el fragmento:

  El 'Stagg Inn' conserva las chimeneas y un horno de pan que se remontan a los tiempos de Robin Hood (casi), pero ha creado una cocina a base de productos de las granjas locales

  El verbo “*ha creado*” (pretérito perfecto compuesto) fue detectado como problemático, alineándose con la categoría esperada del corpus. Este caso ilustra la capacidad del sistema para reconocer tiempos compuestos en contextos con múltiples verbos.

- Falsos Negativos: En 19 casos, el sistema no detectó verbos complejos que deberían haber sido identificados. Un ejemplo representativo es:

  "Hubieras visto hoy a Wilfrido observando las orejas y las patas del lechón en la olla de agua hirviente que Rosa tenía en la cocina."

  Aquí, el verbo "*hubieras visto*" (pretérito pluscuamperfecto del subjuntivo -ra) no fue detectado, posiblemente debido a la complejidad de sintaxis o a la presencia de verbos en gerundio como "*observando"*.

- Falsos Positivos: El sistema identificó incorrectamente 28 fragmentos de la categoría de control "Presente del Indicativo" como problemáticos, lo que representa una tasa de detección falsa del 28% para esta categoría. Un ejemplo es el fragmento:

  "Hay fronteras en las montañas nevadas y hay fronteras en el desierto, en autopistas anónimas y en puentes sobre ríos caudalosos. La frontera entre Inglaterra y el País de Gales es gastronómica, un 'pub' de siglos que se llama el 'Stagg Inn', antigua posada para las diligencias, que ha hecho historia como el primero que recibe una estrella Michelín"

  Aunque filtrado por el verbo "*recibe*" (presente de indicativo), el fragmento contiene el verbo "*ha hecho*" (pretérito perfecto compuesto), que fue detectado como problemático. Este comportamiento se debe a la presencia de verbos adicionales en el corpus, como se explicó previamente, lo que generó falsos positivos al identificar tiempos complejos en fragmentos etiquetados como simples.

- Verdaderos Negativos: En 72 casos, el sistema identificó correctamente fragmentos de la categoría "Presente del Indicativo" como no problemáticos, lo que demuestra su capacidad para descartar formas verbales simples cuando no hay interferencias de otros tiempos verbales.

Estos resultados se pueden ver de forma resumida en la figura 8 que contiene una matriz de confusión

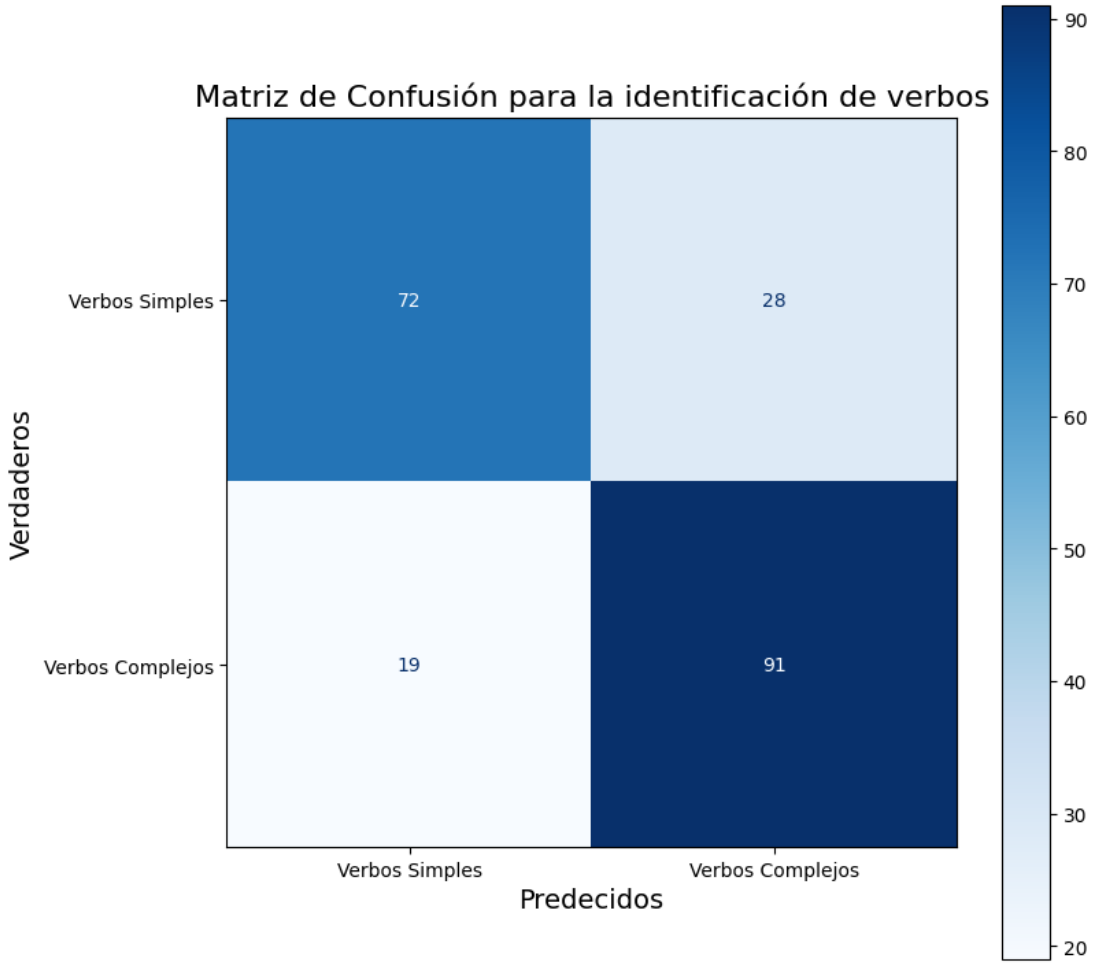

Figura 8 Matriz de confusión de la identificación de verbos

Las categorías con mejor desempeño fueron el condicional compuesto y el futuro compuesto del indicativo con una tasa de detección del 100%. Esto indica que el sistema es altamente efectivo para identificar tiempos compuestos y ciertas formas del condicional cuando las estructuras sintácticas son claras.

Los resultados de las pruebas demuestran que el módulo de identificación es efectivo para detectar la mayoría de los verbos complejos, con una precisión aceptable para este proyecto. La alta tasa de verdaderos positivos y el buen desempeño en tiempos compuestos y condicionales respaldan la elección del modelo de transformadores de spaCy. Sin embargo, los falsos positivos en la categoría de control "Presente del Indicativo" revelan limitaciones importantes. Estas limitaciones se deben, en parte, a la naturaleza del corpus de la RAE, que incluye verbos adicionales no alineados con los filtros aplicados, lo que dificulta la evaluación precisa del sistema. Para mejorar el rendimiento, sería necesario refinar el corpus o reducir los falsos positivos en las categorías de control mediante métodos automatizables más precisos. A pesar de estas áreas de mejora, el módulo cumple con los requisitos del proyecto, proporcionando una base sólida para la posterior adaptación de los verbos complejos en el formato de Lectura Fácil.

3.3.4 Pruebas del módulo de adaptación

Se procesaron un total de 65 fragmentos de texto extraídos del corpus de la RAE, de los cuales 10 correspondieron a la categoría de control «Presente del Indicativo» y 55 a categorías de tiempos verbales complejos. De estos fragmentos, se detectaron problemas en 47 casos. Todos los fragmentos con problemas detectados fueron adaptados, alcanzando una tasa de adaptación del 100% para los casos identificados. La validación manual reveló que 38 de las adaptaciones fueron válidas, representando un 80,85% de las evaluadas, mientras que 9 resultaron no válidas, constituyendo un 19,15%. Entre las adaptaciones válidas, 30 preservaron el significado original, lo que corresponde a un 78,95%, mientras que 8 adaptaciones válidas alteraron el significado, equivalente a un 21,05%. De forma más visual tenemos lo siguiente:

- Muestras totales: 65.
- Muestras con problemas detectados: 47 (72,31%).
- Muestras adaptadas: 47 (100% de los detectados).
- Adaptaciones válidas: 36 (76,60% de las evaluadas).
- Adaptaciones no válidas: 11 (23,40% de las evaluadas).
- Preservación del significado en adaptaciones válidas: 30 (83,33% de las válidas).
- Alteración del significado en adaptaciones válidas: 6 (16,67% de las válidas).

El rendimiento del módulo varió según la categoría de tiempo verbal. Las categorías con mejor desempeño fueron el pretérito perfecto compuesto del indicativo y el pretérito pluscuamperfecto del subjuntivo (-se), logrando un 100% de adaptaciones válidas. Por ejemplo, en el fragmento:

"El 'Stagg Inn' conserva las chimeneas y un horno de pan que se remontan a los tiempos de Robin Hood (casi), pero ha creado una cocina a base de productos de las granjas locales capaz de llamar la atención de los puntillosos inspectores de la guía Michelín"

el verbo "*ha creado*" (pretérito perfecto compuesto) se transformó en "*creó*" (pretérito perfecto simple), manteniendo el significado y cumpliendo con las directrices de Lectura Fácil. En contraste, el futuro compuesto del indicativo y el pretérito pluscuamperfecto del subjuntivo (-ra) presentaron el menor desempeño, con 2 adaptaciones no válidas cada uno. Un caso destacado es el fragmento

"…porque no era 'uno de los suyos', y les anotaba cien carreras en el legendario Lord's casi sin bajarse del autobús. Nieto de un soñador que fue a Australia en busca de oro pero acabó convirtiéndose en granjero, Don Bradman es uno de los grandes deportistas de la historia, y si hubiese sido norteamericano…"

adaptado a

"…porque no era 'uno de los suyos', y les anotaba cien carreras en el legendario Lord's casi sin bajarse del autobús. Nieto de un soñador que fue a Australia en busca de oro pero acabó convirtiéndose en granjero, Don Bradman es uno de los grandes deportistas de la historia, y si había norteamericano…"

donde se puede apreciar un error en la conjugación del verbo principal, resultando en una forma gramaticalmente incorrecta. Esta incidencia es sorprendente ya que no hemos podido encontrar un patrón para este comportamiento erróneo.

Tabla 9 Resumen de los resultados anotados manualmente

Tiempo Verbal	**Detectados**	**Válido**	**No Válido**
Condicional Compuesto	5	4	1
Condicional Simple	5	4	1
Futuro Compuesto del Indicativo	5	3	2
Pretérito Imperfecto del Subjuntivo -se	3	1	2
Pretérito Pluscuamperfecto del Indicativo	4	4	0
Pretérito Pluscuamperfecto del Subjuntivo -ra	5	4	1
Pretérito Pluscuamperfecto del Subjuntivo -se	5	4	1
Pretérito Anterior del Indicativo	5	4	1
Pretérito Perfecto Compuesto del Indicativo	5	5	0
Pretérito Perfecto Compuesto del Subjuntivo	5	3	2
Total	47	36	11

Al realizar las pruebas de este módulo, también se han realizado anotaciones específicas para ciertos casos especiales. Estas anotaciones ofrecen un análisis detallado de los errores y desafíos observados en 18 casos específicos, permitiendo identificar patrones recurrentes que afectan la calidad de las adaptaciones o errores del algoritmo que deben ser corregidos. Dado que múltiples notas repiten el mismo problema, se han agrupado en categorías para poder evaluar la frecuencia de estas ocurrencias. Un listado completo se puede encontrar en el anexo bajo la sección 7.1. A continuación, presentamos una lista de notas comunes:

- Confusión del Verbo Principal: Seis fragmentos muestran errores en la identificación del verbo principal, lo que resulta en adaptaciones incorrectas. Por ejemplo, en "*hubo organizado*" en el texto número 15, se adaptó a "*hubo"*, ignorando "*organizado"* como la forma principal de la acción. Este es posiblemente uno de los errores más graves que ha sido detectado en las últimas fases del proyecto. Posiblemente se deba a un error del algoritmo utilizado y se requeriría una revisión completa del sistema para asegurarse de erradicar un error de esta forma.

- Ambigüedad en la Persona Verbal: Cinco frases presentan dificultades para determinar la persona verbal, generando adaptaciones erróneas. En «Tal vez no me haya explicado bien» (texto 41), se asumió tercera persona en lugar de primera. Similarmente, «Desearía fervientemente [...]» (textos 30 y 61), «Cuánto lo sentiría si hubiere estado [...]» (texto 29), y «habría imaginado» (texto 49) son ambiguas, complicando la selección de la persona correcta. Este es un error común en el procesamiento de lenguaje natural y no es un caso aislado del idioma español. La existencia de frases ambiguas es uno de los mayores retos del PLN y cómo podemos comprobar, hoy en día sigue siendo un problema que spaCy y otras herramientas de PLN intentan solucionar [17].

- Falta de Conservación de Mayúsculas: Tres adaptaciones no respetaron la mayúscula inicial, afectando la presentación. Por ejemplo, «Había sanado» a «sanaba» (texto 21), «Habría llegado» a «llegaba» (índice 37), y «Desearía fervientemente [...]» (texto 61) perdieron la mayúscula, un error menor pero relevante para la consistencia del texto.

- Notas Individuales (3 casos): Tres casos tienen problemas únicos. En «A galope tendido, el perro [...]» (texto 24), se sugiere que, aunque la adaptación es correcta, podrían aplicarse mejoras contextuales alrededor del verbo. En «¡Qué mosca te habrá picado! [...]» (texto 28), se destaca la complejidad de adaptar una frase que usa el futuro para referirse al pasado. En «Deberías avergonzarte de no haberte dado cuenta [...]» (texto 34), la frase no fue bien identificada. Esto se debe a que la frase debería de identificarse como un condicional simple por la aparición de "*Deberías*" al comienzo del texto, pero en su lugar se ha detectado "*haberte dado*" como verbo problemático. Aunque este verbo no cumpla con las guías de Lectura Fácil, no se encuentra dentro del alcance de este Trabajo de Fin de Grado.

Un punto de mejora significativo para el módulo de adaptación sería la integración de un analizador de contexto que permita una selección más precisa del verbo simple al que transformar las formas verbales complejas. Actualmente, el sistema utiliza un enfoque basado en reglas predefinidas, lo que resulta en transformaciones rígidas que no siempre capturan los matices semánticos, especialmente en el caso del condicional simple y formas subjuntivas, donde la elección del verbo simple depende del contexto, la intención del hablante y la estructura de la oración.

4 Conclusiones

Este Trabajo de Fin de Grado ha abordado el desafío de adaptar formas verbales complejas en textos en español, alineándose con las directrices de Lectura Fácil, con el propósito de hacerlos más accesibles para personas con dificultades de comprensión lectora. Durante el desarrollo, se ha implementado un sistema basado en procesamiento de lenguaje natural haciendo uso de la librería spaCy, la cual ha permitido identificar y transformar automáticamente tiempos verbales complejos en formas más simples, buscando preservar el significado original del texto en la medida de lo posible.

El sistema propuesto ha cumplido con los objetivos principales establecidos en la introducción. En primer lugar, se ha logrado identificar automáticamente tiempos verbales complejos como los compuestos, el subjuntivo y el condicional en textos en español. Los resultados de las pruebas del módulo de identificación reflejan la capacidad del sistema para detectar 91 casos de verbos complejos (verdaderos positivos) y descartar adecuadamente 72 casos de formas simples (verdaderos negativos) dándonos una tasa de éxito del 77%. Si tenemos en cuenta los desafíos encontrados en la evaluación del módulo estos resultados nos indican que la elección del modelo de transformadores de spaCy fue acertado para garantizar un buen etiquetado morfosintáctico para este algoritmo. Podemos concluir que el algoritmo de detección desarrollado, aunque no es infalible y aun con los recursos limitados a nuestra disposición, ha sido un éxito para el estándar propuesto en este proyecto.

En segundo lugar, el módulo de adaptación ha transformado con éxito los verbos identificados en formas más simples, siguiendo un conjunto de reglas predefinidas. Los resultados obtenidos en las pruebas de este sistema demuestran la viabilidad del enfoque basado en reglas para cumplir con las directrices de Lectura Fácil. Ejemplos como la transformación de "*ha creado*" a "*creó*" ilustran cómo el sistema logra adaptar tiempos compuestos sin alterar significativamente el contenido. Sin embargo, el 21,05% de adaptaciones válidas que modificaron el significado y el 19,15% de adaptaciones no válidas señalan desafíos pendientes, especialmente en tiempos como el subjuntivo. Principalmente estas pruebas nos indican el mayor desafío en la realización de este módulo. Ciertos errores deben de ser corregidos como la confusión en el verbo principal y el pequeño error de la utilización de mayúsculas cuando un verbo ocupa el inicio de una oración.

Otro desafío importante es la preservación semántica. Aunque el sistema logró mantener el significado en la mayoría de los casos, ciertas transformaciones, como las del subjuntivo pluscuamperfecto a formas de indicativo, alteraron matices temporales o intencionales del texto original. Esto evidencia las limitaciones actuales del PLN para resolver ambigüedades contextuales y sugiere que un enfoque basado únicamente en reglas puede no ser suficiente para abordar todos los casos.

La integración del sistema en una aplicación web, desarrollada con FastAPI y Vite, ha proporcionado una herramienta práctica y accesible que permite interactuar con los módulos de identificación y adaptación sin necesidad de entender las implementaciones que se han llevado a cabo. Esta aplicación no solo visualiza los textos originales y adaptados, sino que también incorpora un sistema de consulta al CORPES, facilitando pruebas rápidas y demostrando la funcionalidad del sistema en un entorno real. La arquitectura modular del diseño asegura que el sistema sea escalable y compatible con proyectos más amplios de adaptación textual, alineándose con iniciativas como el proyecto FACILE.

En términos generales, este trabajo ha demostrado la viabilidad de automatizar la adaptación de formas verbales complejas mediante técnicas de PLN, aportando unos algoritmos que alivia la carga de trabajo manual de editores y adaptadores de textos. Su diseño modular y su enfoque basado en reglas lo convierten en una pieza valiosa para proyectos más ambiciosos, consolidando su relevancia en el ámbito de la accesibilidad lingüística.

Sin embargo, limitaciones como la alteración semántica en transformaciones de subjuntivo, por ejemplo, "Si hubiera sabido" a "sabía", o errores en la identificación de la persona verbal, requieren mejoras para garantizar precisión y maximizar el impacto. Este trabajo sienta las bases para avanzar en la equidad educativa y social mediante el procesamiento de lenguaje natural, alineándose con los objetivos de desarrollo sostenible.

Por otro lado, el sistema desarrollado en este TFG contribuye a la educación de calidad (ODS 4) y reducción de las desigualdades (ODS 10) de la Agenda 2030 al promover la accesibilidad lingüística mediante la adaptación de formas verbales complejas en textos en español. Desde el punto de vista del ODS 4, la herramienta facilita el acceso a materiales educativos para personas con dificultades de comprensión lectora, como aquellas con discapacidades cognitivas apoyando una educación inclusiva aludiendo más concretamente a la meta 4.5. Para el ODS 10, reduce barreras lingüísticas, empoderando a grupos vulnerables y fomentando su inclusión social ayudando a superar la meta 10.2.

En conclusión, el sistema desarrollado representa un avance significativo hacia la automatización de la adaptación textual, aunque requiere refinamientos para alcanzar una precisión y fidelidad óptimas. Los resultados obtenidos validan el potencial del PLN para transformar textos complejos en versiones más comprensibles, al tiempo que destacan la necesidad de seguir explorando estrategias que superen las limitaciones actuales. Este Trabajo de Fin de Grado no solo ha alcanzado sus metas técnicas, sino que también ha contribuido a un objetivo social ayudando a difundir las buenas prácticas de la Lectura Fácil.

5 Trabajos Futuros

En este capítulo se evalúa el impacto potencial de los resultados del TFG en diversos contextos junto con reflexiones sobre posibles mejoras futuras y su integración en proyectos más amplios.

A nivel personal y social, el sistema facilita el acceso a textos para personas con dificultades de comprensión lectora, mejorando su aprendizaje e inclusión, mientras promueve la equidad al reducir barreras lingüísticas; sin embargo, las imprecisiones en las adaptaciones podrían generar malentendidos, afectando la confianza en la herramienta.

En el ámbito cultural y económico, la herramienta democratiza el acceso a obras complejas y reduce los costes de adaptar manualmente los textos para instituciones educativas y ONGs, aunque corre el riesgo de alterar matices estilísticos en textos donde la fidelidad es crucial.

A nivel empresarial y medioambiental, aunque este sistema no este diseñado para su uso de forma empresarial, las metodologías y enseñanzas podrían ser adoptadas por editoriales para cumplir normativas de accesibilidad. Por otro lado, al ser un proyecto de adaptación de textos, la utilidad en el ámbito medioambiental es mínimo ya que no es uno de los objetivos de este proyecto, pero, en defensa de este trabajo, promueve marginalmente el uso de formatos digitales, reduciendo el consumo de papel.

Las decisiones tomadas, como el uso de spaCy por su accesibilidad y la implementación en una aplicación web escalable, reflejan un enfoque hacia la practicidad y la inclusión, maximizando el potencial del sistema en contextos reales. No obstante, las pruebas realizadas revelaron limitaciones que podrían considerarse en futuras iteraciones. Entre ellas destaca la ambigüedad lingüística, como en formas verbales polisémicas, que dificultan una adaptación precisa, y la preservación semántica en ciertas transformaciones muy particulares, donde el significado puede distorsionarse. Asimismo, se observaron errores recurrentes en la identificación del verbo principal en oraciones complejas y en la gestión de mayúsculas, lo que afecta la consistencia y usabilidad del sistema.

Una mejora clave ya mencionada anteriormente para futuras iteraciones del sistema sería la integración de un analizador de contexto para abordar las limitaciones observadas en la preservación semántica y la identificación del verbo principal. Este analizador podría emplear modelos contextuales avanzados para analizar la estructura sintáctica y la intención del hablante, permitiendo una selección más precisa de los tiempos verbales.

Finalmente, el diseño modular del sistema lo hace idóneo para integrarse en proyectos más amplios como FACILE, que buscan combinar múltiples directrices de Lectura Fácil. Su capacidad de colaboración y escalabilidad permite verlo como un componente clave en soluciones más completas, ampliando su impacto en la accesibilidad lingüística y facilitando su adaptación a necesidades diversas en contextos colaborativos.

6 Bibliografía

[1] G. Muñoz, «Lectura fácil: Métodos de redacción y evaluación,» 2012. [En línea]. Available: https://www.plenainclusion.org/sites/default/files/lectura-facil-metodos.pdf.

[2] M. Suárez-Figueroa, I. Diab y E. Ruckhaus, «First steps in the development of a support application for easy-to-read adaptation,» [En línea]. Available: https://doi.org/10.1007/s10209-022-00946-z.

[3] SpaCy, «spacy.io,» 2025. [En línea]. Available: https://spacy.io.

[4] FastAPI, «fastapi.tiangolo.com,» 2025. [En línea]. Available: https://fastapi.tiangolo.com/.

[5] Vite, «vite.dev,» 2025. [En línea]. Available: https://vite.dev/.

[6] «Making information accessible for all | european blind union,» [En línea]. Available: https://www.euroblind.org/publications-and-resources/making-information-accessible-all#Text.

[7] AENOR: Asociación Española de Normalización y Certificación, 2008. [En línea]. Available: https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0060036.

[8] S. J. Mielke, Z. Alyafeai, E. Salesky, C. Raffel, M. Dey y M. Gallé, «Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP,» [En línea]. Available: https://arxiv.org/pdf/2112.10508.

[9] R. Cotterell y G. Heigold, «Cross-lingual, character-level neural morphological tagging,» [En línea]. Available: https://arxiv.org/pdf/1708.09157.

[10] «Universal POS tags,» [En línea]. Available: https://universaldependencies.org/u/pos/.

[11] «Universal features,» [En línea]. Available: https://universaldependencies.org/u/feat/index.html.

[12] Python, «https://www.python.org/python.org,» 2025. [En línea]. Available: https://www.python.org/.

[13] J. Antona Palacios, «Lectura Fácil: perífrasis verbales 2.0,» Junio 2022. [En línea]. Available: https://oa.upm.es/71755/.

[14] A. Ming Díaz, «Lectura Fácil: perífrasis verbales,» Junio 2022. [En línea]. Available: https://oa.upm.es/71605/.

[15] RAE, «www.rae.es,» [En línea]. Available: https://www.rae.es/corpes/.

[16] RAE-API, «rae-api.com,» [En línea]. Available: https://rae-api.com/.

[17] A. Yadav, A. Patel y M. Shah, «A comprehensive review on resolving ambiguities in natural language processing,» 2021. [En línea]. Available: https://doi.org/10.1016/j.aiopen.2021.05.001.

7 Anexos

7.1 Notas tomadas en el módulo de adaptación

15	Un ejemplo de menú criollo es el que ofrece el Venezuela Restaurante de la ciudad de La Paz, en el festival culinario que hubo organizado.	"Hubo organizado" lo transforma a "Hubo" Debe haber algún problema con el reconocimiento del verbo principal
21	-Lo curioso es que a los pocos minutos su piel no guardaba memoria de la herida. Había sanado.	Aunque este bien adaptado, no se conserva la mayúscula "Había sanado" se transforma en "sanaba"
24	A galope tendido, el perro, hecho una sombra, corre como si hubiese escapado de una prisión sobrenatural y, pateando el aire, tratase de quitarse del pescuezo una flecha invisible que le impide resollar.	Aunque bien transformado, se podría aplicar mejores adaptaciones alrededor del verbo
27	-Me pregunto qué habrán ido a hacer, y en último término, qué hacemos aquí.	Mismo fallo de verbo principal: "habrán ido" a "habrán"
28	-¡Qué mosca te habrá picado! Claro que no. ¿Tengo cara de periodista?	Esta frase es complicada ya que habla sobre el pasado usando el futuro
29	— Cuánto lo sentiría si hubiere estado en ese momento.	Se ha equivocado poniendo tercera persona en vez de primera por ser una frase ambigua
30	Desearía fervientemente que la banda sonora y musical de "Edipo Asesor" tuviese el registro y la envergadura de la música del "nuevo cine alemán".	Dos problemas, es una frase ambigua (3ª o 1ª persona) y No pone mayúscula
34	Francisco: Deberías avergonzarte de no haberte dado cuenta, tú... que me pariste.	Esta frase no está bien identificada
37	¿Habría llegado la hora de anunciar que entre «representar a» y «representar para» había tanta semejanza y confluencia de acciones y resultados que distraerse en diferenciarlos ya no vale la pena?, se preguntaba. Tomando al pie de la letra las	Otra vez la mayúscula no puesta

	recomendaciones del documento europeo que ya cumplía dos años circulando	
38	, un tal Tim Matthews, encontró las notas al hacer limpieza en los sótanos de su casa de campo y encontrarse con viejos álbumes de familia que llevaban más de un siglo y medio criando polvo. '¿Qué tenemos para cenar esta noche?' fue ya un best-seller cuando se publicó bajo pseudónimo, pero habría sido una auténtica sensación de saberse que el autor era nada menos que Charles Dickens', comenta su biógrafo Peter Ackroyd.	Vuelve a confundir el verbo principal
41	Karlos.- Querrá decir a dónde. Tal vez no me haya explicado bien. Quiero decir pasar en el sentido de entrar.	Frase ambigua para un ordenador, pero parece ser primera persona en vez de tercera
42	Ah, no, no, gracias. No es que se me haya caído, es que la he tirado.	Debería de usar otra forma verbal como el pretérito perfecto simple
45	, porque no era 'uno de los suyos', y les anotaba cien carreras en el legendario Lord's casi sin bajarse del autobús. Nieto de un soñador que fue a Australia en busca de oro pero acabó convirtiéndose en granjero, Don Bradman es uno de los grandes deportistas de la historia, y si hubiese sido norteamericano habría películas y libros sobre él, y calles que llevasen su nombre. Su promedio de noventa y nueve carreras por partido internacional es el equivalente de cien goles por temporada en una liga de fútbol como la española o la italiana. Casi inimaginable.	Vuelve a confundir el verbo principal
49	Pero jamás las habría imaginado juntas, si ella no hubiese usado la palabra «sentimientos».	Vuelve a ser un caso ambiguo
50	Álvaro — Si yo hubiera sabido.	Esta frase es difícil de adaptar a Lectura Fácil
53	-Me hubieran sido más cómodos los originales...	Vuelve a confundir el verbo principal
61	Desearía fervientemente que la banda sonora y musical de "Edipo Asesor" tuviese el registro y la envergadura de la música del "nuevo cine alemán".	Frase ambigua y mayúscula

63	anotaba cien carreras en el legendario Lord's casi sin bajarse del autobús. Nieto de un soñador que fue a Australia en busca de oro pero acabó convirtiéndose en granjero, Don Bradman es uno de los grandes deportistas de la historia, y si hubiese sido norteamericano habría películas y libros sobre él, y calles que llevasen su nombre. Su promedio de noventa y nueve carreras por partido internacional es el equivalente de cien goles por temporada en una liga de fútbol como la española o la italiana. Casi inimaginable.	Verbo principal equivocado

Recibo digital

Este recibo confirma quesu trabajo ha sido recibido por Turnitin. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega:	ALEJANDRO PERIS ACEDO
Título del ejercicio:	Turnitin Memoria Final
Título de la entrega:	TFG Alejandro Peris.pdf
Nombre del archivo:	26370_ALEJANDRO_PERIS_ACEDO_TFG_Alejandro_Peris_83714...
Tamaño del archivo:	571.94K
Total páginas:	45
Total de palabras:	11,119
Total de caracteres:	63,829
Fecha de entrega:	03-jun.-2025 07:15p. m. (UTC+0200)
Identificador de la entrega:	2691440025

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Grado en Matemáticas e Informática

Trabajo Fin de Grado

Lectura Fácil: Tiempos Verbales

Autor: Alejandro Peris Acedo
Tutora: María del Carmen Suárez de Figueroa Baonza
Cotutor: Isam Diab Lozano

Madrid, junio 2025

Derechos de autor 2025 Turnitin. Todos los derechos reservados.

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Wed Jun 04 18:48:45 CEST 2025
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)