Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Doble Grado en Ingeniería Informática + ADE

Trabajo Fin de Grado

Lectura Fácil: Formato de diálogos en microcuentos

Autor: Roberto José Peris Acedo
Tutor(a): María del Carmen Suárez de Figueroa Baonza
Cotutor(a): Isam Diab Lozano

Madrid, Junio - 2023

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
Doble Grado en Ingeniería Informática + ADE

Título: Lectura Fácil: Formato de diálogos en microcuentos

Junio - 2023

Autor: Roberto José Peris Acedo
Tutor: 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 consiste en unas directrices para adaptar los textos a un lenguaje más fácilmente comprensible. Con esto se busca obtener textos claros y fáciles de entender para permitir su accesibilidad a todo tipo de persona con algún tipo de dificultad lectora.

El objetivo principal de este trabajo de fin de grado es investigar la posibilidad de convertir los diálogos de los micro-cuentos en una versión más clara. Esta conversión se basa principalmente en una reorganización del diálogo para facilitar su comprensión.

Para la investigación propuesta, se van a identificar dos posibles enfoques. Un primer enfoque que engloba todos los diálogos con incisos y con un orador claro. El otro enfoque consiste en el resto de casos donde los diálogos no tienen un inciso o no se menciona quien es el orador. Para ambos enfoques se van a usar una base de datos de microrrelatos sencillos, destacando “El niño del pijama de rayas”.

Durante el trabajo se mostrarán los resultados obtenidos y los casos de prueba para ver la posible viabilidad del proyecto. Además se expondrá cuales son los casos que mejor funcionan para su posible uso futuro.

Abstract

Easy-to-read consists of guidelines to adapt texts into more understandable language. With this, it seeks to obtain clear and easy to comprehend texts that enable its accessibility to all kind of people with some type of reading difficulty.

The main objective of this final degree project is to investigate the possibility to convert micro-story dialogues into a clearer version. This conversion is based mainly on a reorganization of the dialogue to ease its comprehension.

For the purposed research, two possible approaches will be identify. The first approach covers all dialogues with dialogue tag and with a clear speaker. The other approach consist of the rest of cases where there is no dialogue tag nor speaker. An easy micro-stories data base will be used in both approaches, highlighting “The boy in striped pajamas”.

Obtained results and test cases will be shown during the project to analyse its viability. Also, cases that work the best will be shown for its possible use in the future.

Tabla de contenidos

Índice de figuras

Capítulo 1

Introducción

La metodología de **Lectura fácil** es un conjunto de pautas y recomendaciones que se aplican a los textos con el fin de facilitar su comprensión sin perder información en el proceso. El principal objetivo que tiene es ayudar a la gente con discapacidad cognitiva a que entiendan de una forma más sencilla los textos, pero también puede ser útil para personas con problemas de aprendizaje, personas mayores y personas con problemas de visión entre otros.

Estas guías y pautas son recientes, no fue hasta 1997 cuando se publicó el primer documento sobre lectura fácil por la *Comisión Europea*. Este documento se llamaba *Making Information Accessible to All: Recommendations for the Presentation of Written Information for Blind and Partially Sighted People* [1], aquí se incluían las pautas originales de lectura fácil. Este informe originalmente estaba destinado para la población con cualquier impedimento visual (visión borrosa, ceguera nocturna y otros impedimentos), pero a su vez también incluía recomendaciones para hacer más accesible la información para personas con discapacidad cognitiva o con habilidades limitadas de alfabetización. A raíz de esta publicación, las pautas se han ido actualizando a lo largo de los años hasta llegar al presente donde aún no se ha publicado un documento oficial que las recoja todas. La documentación española más moderna, con la que se esta trabajando, es *Lectura fácil: Métodos de redacción y evaluación* [2], elaborado por el Gobierno de España en 2010 y actualizado en 2018.

Este TFG se realiza en el contexto de la línea de investigación Inteligencia Artificial Aplicada: Mejora de la Accesibilidad Cognitiva, cuyo objetivo es utilizar técnicas de Inteligencia Artificial para mejorar la Accesibilidad Cognitiva de Textos. Este trabajo trata una función en especifico para la conversión de textos a Lectura Fácil. La función en cuestión es la conversión de diálogos a una versión de lectura fácil en microcuentos[1]. Los diálogos tienen una estructura muy especifica y a la vez compleja debido a la necesidad del lector de sobreentender la identidad de los interlocutores. Por este motivo, la técnica de Inteligencia Artificial que se va a usar es el **PLN** (Procesamiento del Lenguaje Natural), que es la

[1]El término microcuento se va a usar como término que englobe otros tipos de textos como relatos cortos, narrativa breve, microcuentos y literatura breve entre otros documentos.

rama de la inteligencia artificial que se encarga de procesar el lenguaje natural.

Este proyecto forma parte del ámbito del procesamiento del lenguaje natural (PLN o NLP en inglés), el cual es una disciplina de la Inteligencia Artificial que se dedica a diseñar programas de ordenador que puedan realizar diversas tareas relacionadas con el lenguaje humano (Jurafsky y Martin, 2008) [3]. Esta área de estudio busca desarrollar recursos, técnicas y métodos computacionales que permitan a las máquinas "entender" y "usar" el lenguaje humano. Surgió en los años 50 cuando se comenzó a desarrollar los sistemas de traducción automáticos por IBM. El PLN es una técnica que se ha desarrollado mucho en los últimos años, y que se ha aplicado a muchos campos, como la traducción automática, la detección de spam, y la clasificación de textos. Está presente en muchos ejemplos de nuestra vida cotidiana, como puede ser un asistente virtual como Siri o Alexa, que nos ayuda a realizar tareas como buscar información en internet, o pedir un taxi.

El objetivo principal de este trabajo es la investigación de la posibilidad de implementar una función que permita convertir diálogos a una versión de lectura fácil en microcuentos. Dado que es un proceso de investigación, no se espera que el resultado sea perfecto, sino que se pretende que sea una aproximación a la solución del problema.

Capítulo 2

Estado de la Cuestión

Previamente al comienzo del desarrollo del proyecto, es necesario realizar una investigación sobre la tecnología que se va a usar, así como sobre las herramientas que se van a utilizar para el desarrollo del proyecto. Además, se van a estudiar los trabajos ya existentes sobre la automatización de las pautas de lectura fácil.

2.1. Trabajos Previos

Antes de comenzar con el desarrollo del trabajo se ha realizado una investigación de la Metodología de Lectura Fácil y de las herramientas existentes usadas para el Procesamiento del Lenguaje Natural. Es necesario conocer la importancia y el impacto que pueden llegar a tener estos documentos para abordar correctamente el TFG. Con respecto a la Metodología de Lectura Fácil existen unas pautas a seguir para producir diálogos claros que se van a seguir en el trabajo actual.

Para añadir a esta investigación, se han analizado los trabajos de fin de grados anteriores que tenían relación con la Lectura Fácil. Al tratar una funcionalidad tan única como los diálogos, en este TFG no se pueden aprovechar algoritmos específicos ni casos de pruebas. A pesar de ello, se puede obtener información relevante con respecto al uso de las herramientas de PLN y otras reglas de la Lectura Fácil.

2.2. Procesamiento del Lenguaje Natural

Para comenzar, el proyecto se basa en una rama de la Inteligencia Artificial, más concretamente en el Procesamiento del Lenguaje Natural (PLN). El PLN es un área de investigación y aplicación que explora cómo los ordenadores pueden entender y usar el lenguaje natural de forma escrita o hablada. El PLN es una técnica de la inteligencia artificial que busca ser utilizada principalmente para solventar las diferencias de dialectos, idiomas y culturas, entre otras cosas [4]. Para poder realizar esto, es necesario desarrollar algoritmos y modelos computacionales que permitan a estas máquinas el poder analizar, procesar y

generar lenguaje natural. El PLN tiene varias aplicaciones, entre ellas se pueden encontrar los famosos *chatbots* y los asistentes virtuales que interactúan con los usuarios a través de la voz como por ejemplo Alexa o Siri [5].

2.2.1. Tareas del PLN

El PLN esta dividido en una serie de tareas, las cuales están bien diferenciadas y muchas de ellas son necesarias para la realización del trabajo. A pesar de que las tareas no tienen un orden fijo se pueden estructurar a través de lo que se conoce como "pipeline". Una pipeline es una serie de eventos linealmente ordenados que pretende construir un software PLN [6]. Un ejemplo de pipeline es el caso de **CLAMP**, un software clínico que usa el PLN y tiene la siguiente pipeline:

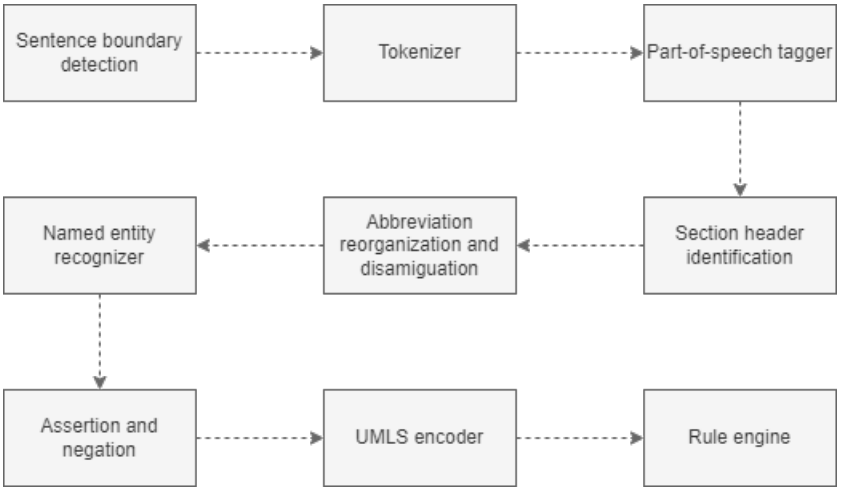

Figura 2.1: *Pipeline CLAMP*

De todas las tareas existentes en el PLN, las más relevantes para este trabajo son:

- Preprocesado del texto
- Tokenizador
- Etiquetado
- Reconocimiento de entidades
- Análisis

A pesar de no tener un orden fijo, es cierto que se suele efectuar algunas tareas antes que otras. Un ejemplo es realizar la tarea de preprocesado del texto previamente a cualquier tarea de análisis.

2.2.1.1. Tarea de Preprocesado del texto

El objetivo principal de esta tarea es "limpiar" el texto, eliminando, entre otros, caracteres especiales, signos de puntuación y las conocidas como *stopwords* (pa-

labras comunes en un idioma que no aportan información, como preposiciones o conjunciones) [7]. La principal razón para esta conversión es que el algoritmo no pueda interpretar estos caracteres como parte del texto, ya que no aportan información relevante, además de que mejoran la calidad de los resultados. Esta tarea varía dependiendo de la información que se pretenda obtener. Ejemplo con stopwords:

> Original: Decía siempre una de las amigas antipáticas de su hermana.
> Procesado: Decía siempre una amigas antipáticas hermana.

2.2.1.2. Tarea de Tokenizado

En esta tarea se divide el texto en unidades más pequeñas, llamadas tokens. Estos tokens pueden ser palabras, frases, símbolos, y otras opciones. En ocasiones, puede resultar conveniente mantener los signos de puntuación y otros símbolos como tokens, ya que proporcionan información útil al texto. El tokenizado es una tarea muy importante, ya que permite al algoritmo entender el texto. Existen diversas librerías y recursos externos del PLN para tokenizar [8]. Ejemplo:

> Original: Decía siempre una de las amigas antipáticas de su hermana.
> Tokenizado: [Decía, siempre, una, de, las, amigas, antipáticas, de, su, hermana]

2.2.1.3. Tarea de Etiquetado

Esta tarea consiste en asignar una etiqueta PoS (Part of Speech), con información morfosintáctica a cada token [9]. Estas etiquetas pueden ser de diferentes tipos, como por ejemplo, la categoría gramatical de la palabra, el tipo de palabra (sustantivo, verbo, adjetivo), el tipo de entidad (persona, lugar, organización), y otros casos. Su objetivo es organizar cada uno de los tokens y entender la estructura gramatical del texto dado. Esto facilitará futuras tareas como los análisis sintácticos o semánticos. Ejemplo:

> Original: Decía siempre una de las amigas antipáticas de su hermana.
> Tokenizado: [Decía, siempre, una, de, las, amigas, antipáticas, de, su, hermana]
> Etiquetado:
>
> > Decía - Verbo
> > siempre - Adverbio
> > una - Pronombre
> > de - Preposición
> > las - Determinante
> > amigas - Sustantivo
> > antipáticas - Adjetivo
> > de - Preposición
> > su - Determinante
> > hermana - Sustantivo

2.2.1.4. Tarea de Reconocimiento de entidades

La tarea de Reconocimiento de entidades (Named Entity Recognition NER) consiste en identificar y clasificar las entidades nominales que aparecen en el texto [9]. Estas entidades pueden ser elementos como personas, lugares u organizaciones. La razón por la cual se realiza esta fase es debido a que permite una mayor rapidez a la hora de responder preguntas o resumir textos. Por ejemplo, si se le pregunta al algoritmo *¿Quién es el presidente de España?*, este podrá responder *El presidente de España es Pedro Sánchez*. Si no se realizara esta fase, el algoritmo tendría que buscar en todo el texto la palabra *presidente* y *España* para poder responder a la pregunta.

2.2.2. Tarea de Análisis

Esta tarea se basa en analizar los textos. Estos análisis pueden ser de varios tipos pero los principales son análisis semántico, análisis de sentimientos y análisis de dependencia. La principal función de estos análisis es comprender el significado de las palabras en su conjunto, mas allá del significado literal de cada palabra.

El análisis semántico se compone de dos procedimientos. El primero es un análisis léxico donde la máquina analiza el significado individual de cada palabra. El segundo es un análisis composicional, donde da un sentido a la frase en base al significado de las palabras [5].

El análisis de sentimientos es un reto, ya que tiene una gran connotación subjetiva que es difícil de reflejar en un texto. Se analizan principalmente tres factores: las expresiones de sentimientos, la polaridad y la fuerza de las expresiones y su relación con el sujeto [10].

Por último, el análisis de dependencia consiste en realizar un análisis sintáctico y examinar las dependencias existentes entre las palabras de una oración. Un ejemplo del mismo es la siguiente figura 2.2:

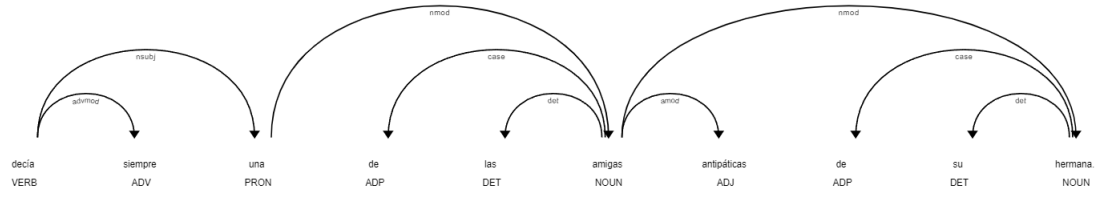

Figura 2.2: *Ejemplo análisis dependencia*

2.3. Lectura Fácil

Como se ha expuesto anteriormente, las pautas de la metodología de Lectura Fácil son muy modernas, y se originaron en 1997. Por ello, estas pautas están

en constante evolución y mejorándose aun a día de hoy. Lo que se ha conseguido con estas pautas es su implementación en algunos campos como puede ser el campo gubernamental, el campo del negocio o el campo educativo entre otros. Más importante aun, se ha buscado su implementación en los medios de transmisión de información. En el caso de la literatura se ha buscado convertir los clásicos como "El principito" a una versión de lectura fácil [11].

Al ser relativamente nuevo, la forma de implementarlo a los textos ha sido muy variada y sigue en constante cambio. Aun así, hoy en día la forma estandarizada de aplicarlo son las pautas de la metodología (estándar de la lectura fácil). Esto se realiza de forma manual en cada texto y no se ha automatizado a gran escala debido a que aún no existe un sistema que lo realice. A pesar de esto, existen empresas que buscan formas para añadir estas pautas a sus servicios de forma automatizada. Para este proceso lo que está en el punto de mira es la Inteligencia Artificial debido principalmente a su creciente popularidad. Existe una iniciativa que está buscando incorporar la Lectura Fácil en el navegador. Este proyecto es conocido como Web Accessibility Iniative (WAI) [12]. Su objetivo es conseguir implementar de alguna forma la Lectura Fácil en el navegador. Para ello, ha realizado estudios y creado una serie de pautas para la accesibilidad, estas pautas se las conoce como Pautas de Accesibilidad para el Contenido Web (WCAG). Actualmente están en la versión 2.1, que fue actualizada en 2018. Además de realizar pautas, también crean herramientas y otros recursos para añadir al navegador para acercarse a este objetivo. Algún ejemplo de herramientas disponibles de este proyecto son el Atester o el a11yTools para navegador [13].

Como se ha mencionado con anterioridad, existen unas pautas de la metodología de lectura fácil para los diálogos [2]. Es importante seguir estas reglas para poder crear un diálogo lo más claro posible. Las pautas en cuestión son las siguientes:

- "Se puede utilizar el guión o la raya para iniciar los diálogos"
- "Introducir el nombre de la persona que habla a continuación de forma narrada, para evitar subordinadas"
- "Redactar los monólogos, monólogos interiores y diálogos como acotaciones teatrales, anteponiendo el nombre del personaje en mayúsculas antes de las palabras que exprese. De este modo, se mantiene el interés del lector y se ordenan las intervenciones de los personajes. El formato teatral de diálogo es fácil de seguir, da más dinamismo a la narración y permite al lector imaginarse la acción como una pieza teatral o una película"
- "Sangrar los textos de los diálogos para diferenciarlos de la narración."

2.4. Herramientas

Las principales herramientas que se usan a día de hoy en el PLN son las siguientes: NLTK[1] , spaCy[2], Stanford NLP[3], Apache OpenPLN[4] y Gate[5] [14]. La herramienta a usar en este TFG será **spaCy**, ya que es la idónea para la tarea. Esta herramienta fue creada por Matthew Honnibal y Ines Montani y se pretendía que fuese rápida y eficiente.

spaCy es una librería de Python de código abierto que permite desarrollar aplicaciones basadas en el PLN. Tiene varias funcionalidades, entre ellas el análisis de textos y la tokenización. Tiene tres tipos de objetos: Token, Span y Doc. El Token es un objeto que representa una palabra o símbolo. El Span es un objeto que representa una secuencia de Tokens. El Doc es un objeto que representa una secuencia de Tokens y Spans [15].

Lo que hace destacar a spaCy es su eficiente algoritmo de tokenización, que es capaz de manipular grandes cantidades de textos. Además, incluye modelos preentrenados para varios idiomas, entre ellas el español. Además de la tokenización, destaca en su etiquetado, el cual se basa en el framework llamado *Universal Dependencies* [16]. Este framework comunitario se encarga de etiquetar palabras en estructuras sintácticas. spaCy no solo es capaz de realizar etiquetados sintácticos, sino que también es capaz de analizar la morfología y la semántica de las oraciones.

Además de estas funcionalidades existen otras muchas como el análisis de *chunks* o el árbol de análisis de dependencias, las cuales se explicarán posteriormente. Todas ellas posibilitarán el desarrollo del trabajo para obtener la mayor cantidad de resultados posibles.

[1] NLTK: `https://www.nltk.org`
[2] spaCy: `https://spaCy.io`
[3] Stanford NLP: `https://nlp.stanford.edu`
[4] Apache OpenPLN: `https://opennlp.apache.org`
[5] Gate: `https://gate.ac.uk`

Capítulo 3

Desarrollo

3.1. Planteamiento del problema

Existe una gran variedad de idiomas en el mundo, y cada uno de ellos tiene sus propias reglas gramaticales. El idioma más usado para el PLN es el ingles, a pesar de ello, cada vez más idiomas están siendo adaptados para su uso. Uno de estos idiomas es el español de España, el cual se va a utilizar en este trabajo. Más concretamente, el caso de estudio a tratar es con los diálogos. No existe una metodología oficial para escribir diálogo, ya que cada autor los escribe de una forma diferente. Debido a esto, es necesario usar una estructura en concreto (la más estandarizada) y trabajar en base a ella. Esta estructura contiene dos elementos:

- **Parlamento:** Intervenciones del propio personaje.
- **Inciso:** Aclaraciones que hace el narrador.

Ejemplo: *—Está haciendo las maletas —respondió la Madre.* [17]

En este caso, el parlamento es "Esta haciendo las maletas" y el inciso es "respondió la Madre". Para identificar los diálogos se va a dar por hecho que se introducen por un guion y pueden presentar otro guión si tienen incisos. Además, se va a dar por hecho que los diálogos siempre empiezan en una linea nueva.

Existen varias clases de diálogos, literarios, en relatos y teatrales. Los objetivos son los diálogos en relatos, los cuales se subdividen en estilo directo, indirecto y resumido. En el estilo directo, el narrador deja hablar directamente al personaje. En el estilo indirecto, el narrador inserta literalmente las palabras de los personajes. En el estilo resumido, el narrador abrevia el diálogo haciendo un resumen [18]. En este caso, se va a usar el estilo directo, ya que es el que más se usa en los relatos.

El problema que se pretende abarcar en este TFG es el de convertir los diálogos de los microcuentos en su versión de Lectura Fácil. Un ejemplo es el siguiente:

[INCORRECTO]

—¡Bruno, te he dicho que subas y deshagas las maletas ahora mismo! —le ordenó la Madre.

[CORRECTO]
La Madre le ordenó:
¡Bruno, te he dicho que subas y deshagas las maletas ahora mismo!

La premisa parece sencilla, pero requiere de un análisis profundo de la estructura de los diálogos para poder llevar a cabo la tarea. La principal razón para esto es debido a la gran variedad de estructuras que pueden tener los diálogos. Para realizar este proyecto se ha planteado un enfoque para poder abordar el planteamiento.

Primero, es necesario recopilar un *corpus* (Conjunto de datos para realizar la investigación). Para una aproximación inicial, los datos se pretendían que fuesen del CREA (Corpus de Referencia del Español Actual) [1], más concretamente de la sección literaria. Sin embargo, debido a la complejidad de estos diálogos, se ha optado por cambiar de corpus. Por lo tanto, el corpus a usar será una colección de microcuentos y de algún libro sencillo, como por ejemplo *El Principito* o *El niño del pijama de rayas*. Estos diálogos son mucho más sencillos y se aproximan más a la idea original del proyecto, la de convertir diálogos provenientes de los microcuentos.

Tras encontrar un corpus base con el que comenzar a trabajar (colección de 50 cuentos populares), hay que realizar las distintas tareas de procesamiento de lenguaje natural. Aparte de las tareas comunes como puede ser tokenizar y reconocimiento de entidades, hay que elegir el modelo a utilizar. Al tratarse de un problema en Español, la lista de modelos a usar se reduce a los cuatro que ofrece spaCy: Modelo pequeño en español, modelo mediano en español, modelo grande en español y modelo de transformadores en español. Se va a usar el modelo grande en español, debido a que el pequeño tiene muy pocos datos como para dar unos valores precisos. El modelo de transformador es muy preciso, pero requiere de más capacidad de cómputo. Se ha realizado una comparación posterior en *Casos de prueba* para explicar por que se ha decidido usar el modelo grande sobre el mediano.

La siguiente tarea a realizar, es la de detectar las estructuras de diálogos. Para esta tarea se va a usar un enfoque declarativo mediante un método basado en reglas. Tras estos preparativos, se convertirían estas estructuras a un estilo teatral de Lectura Fácil. Por último, se realizaría una comparación entre los diálogos originales y los diálogos convertidos.

[1] CREA: `https://www.rae.es/banco-de-datos/crea`

3.2. Implementación

3.2.1. Detección de diálogos

Lo primero a implementar son las reglas para detectar los diálogos en los textos del corpus. El corpus para este trabajo se encuentra en formato txt, ya que los trabajos previos a este TFG insertan los textos de esta forma.

Para identificar los diálogos del corpus, es necesario identificar los elementos comunes que suelen tener los diálogos. Estos elementos principalmente son la raya (—) [19] y el guión (–) [20]. Estos elementos aparecen al principio del parlamento y del inciso, pudiendo haber un tercero al final del inciso para continuar el parlamento. Además de esto, los diálogos que se van a tener en cuenta son los que comienzan en una nueva línea. Por lo tanto, el primer paso de implementación es recorrer todos los párrafos del texto hasta sacar todos los diálogos existentes. Para detectar qué es un diálogo y qué no, se aplica un algoritmo que revisa si la oración comienza con una raya o guión.

Existen dos casos a la hora de detectar diálogos. Si el diálogo detectado cuenta con dos o tres guiones se trata de un diálogo con inciso hechos por el narrador, si tiene un único guión se trata de un diálogo sin inciso. Cada uno de los casos requieren de una investigación individualizada, por lo que resultarán en dos algoritmos distintos.

3.2.2. Diálogos con incisos

En este primer caso, los diálogos cuentan con un inciso, en el cual debe estar incluido el orador del diálogo. En este inciso existe un elemento importante, la anáfora. Una anáfora es una relación de identidad que se establece entre un elemento gramatical y una palabra o grupo de palabras nombrados antes en el discurso [21]. En el caso de los diálogos, en el inciso la anáfora existente es la mención al orador. Un ejemplo claro del caso es el siguiente:

> —¡Bruno, te he dicho que deshagas las maletas ahora mismo! —le ordenó la Madre.

Tras aplicarle el algoritmo de conversión de diálogo debería quedar de la siguiente manera:

> La Madre le ordenó:
> ¡Bruno, te he dicho que subas y deshagas las maletas ahora mismo!

3.2.2.1. Enfoque

Para llegar a la solución de este problema, se ha decidido tener una serie de pasos. Gracias a este enfoque, se puede abordar el problema de una forma más ordenada y clara, con el fin de evitar correcciones innecesarias o futuros problemas.

El primer paso en cuestión es obtener el **interlocutor** del diálogo, es decir, el personaje que esté hablando. Para ello se ha pensado en varias formas de hacerlo. Aunque antes de ello, hay que hacer una reflexión sobre qué es este interlocutor para spaCy y, más concretamente para el modelo que se está usando, el modelo grande en español. Como se ha expuesto anteriormente, para spaCy, el interlocutor es un token que tendrá distintas características dependiendo de la oración. Lo que se sabe con certeza es que es un **pronombre** o un **sustantivo**, el cual puede ir precedido de un determinante y puede ir sucedido de algún adjetivo o algún adverbio. En el ejemplo anterior podemos ver que el sujeto esta precedido de un determinante:

> —¡Bruno, te he dicho que deshagas las maletas ahora mismo! —le ordenó la madre.
>
> Sujeto: La madre

A nivel sintáctico el interlocutor puede ser nombre común, como *madre* o nombre propio, como *Bruno*. A nivel semántico puede ser el sujeto, un complemento directo o un complemento indirecto. El caso más común es que un interlocutor sea un nombre propio, que actué como sujeto y que esté precedido por un determinante. Existe algún caso donde el sujeto es un adjetivo sustantivado en vez de un sustantivo. En caso de que no exista un sujeto se le categorizará como "sujeto omitido".

Tras identificar al interlocutor, el ultimo paso es ordenar la frase para colocar el inciso antes del parlamento. Para ello, hay que reconstruir el inciso colocando el sujeto primero, dejando el verbo en ultimo lugar y en el medio el resto de aclaraciones extras del inciso.

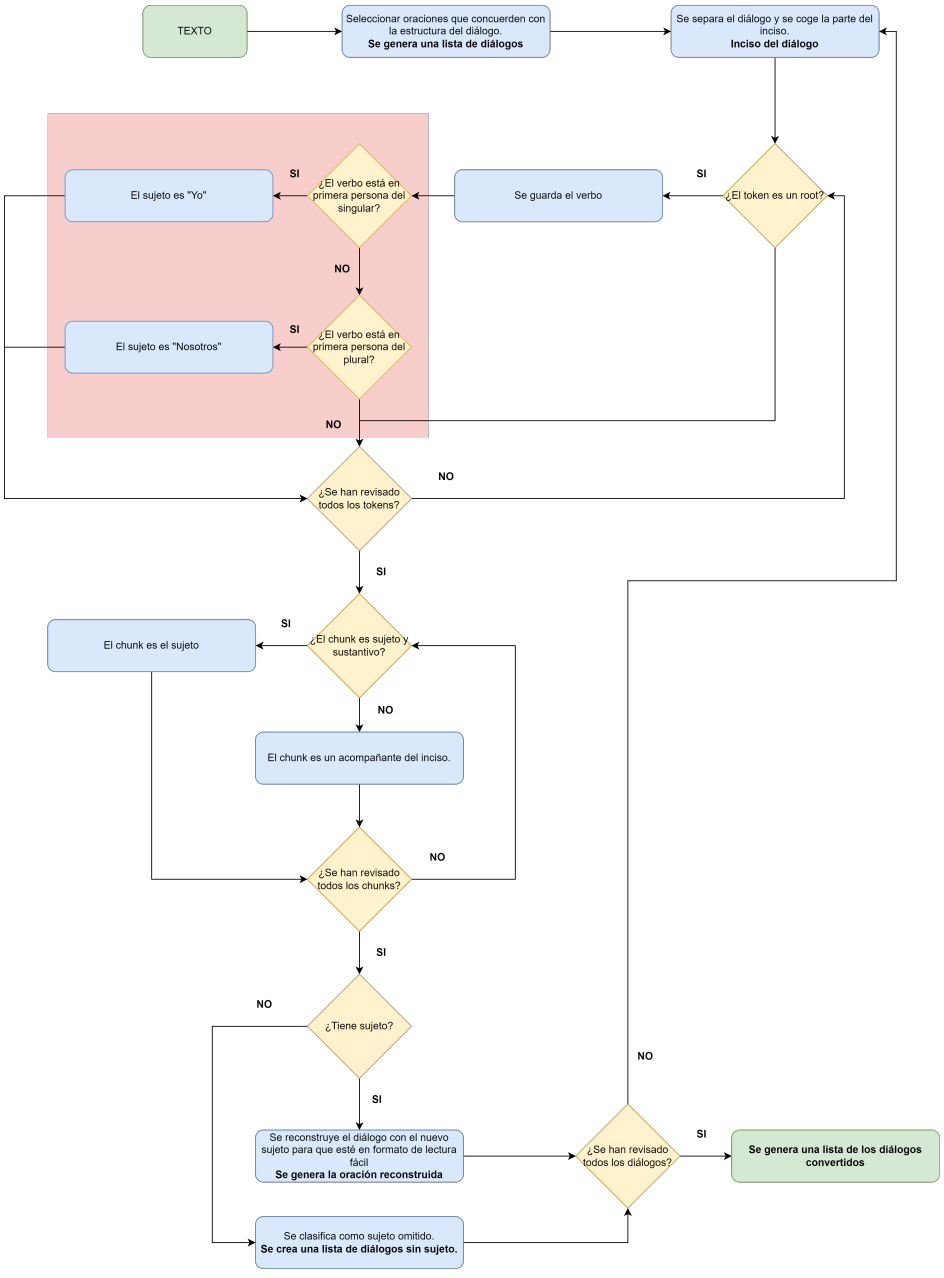

Figura 3.1: *Diagrama de flujo del caso 1*

En la figura 3.1 se puede apreciar de una forma más gráfica el algoritmo que se ha usado. La sección en rojo pertenece al segundo caso a estudiar que se explicará más adelante, pero se incluye en este algoritmo.

3.2.2.2. Implementación final

Para implementar el primer paso, se va a necesitar realizar un bucle recorriendo todos los diálogos que se han detectado anteriormente. Para que no haya problemas de formatos, se transforman los guiones al formato de guión largo. Posteriormente para optimizar el proceso, se va a analizar únicamente la parte del inciso en busca del sujeto. Para ello, se divide el diálogo y se selecciona el inciso.

Una vez cogido el inciso se creará un nuevo bucle por cada token del inciso. Estos tokens son proporcionados por spaCy. La funcionalidad de este bucle es principalmente obtener el verbo del inciso. En versiones anteriores del algoritmo, el bucle tenia dos funcionalidades más. Una primera que consistía en guardar en un diccionario todos los posibles interlocutores existentes, entre ellos sujetos, complementos directos e indirectos y que sean o sustantivos o nombres propios. La segunda funcionalidad consistía en guardar en una lista el resto de elementos del inciso que acompañen al sujeto. Con estas dos funcionalidades extras se pretendía fusionarse con el fin de crear el inciso final. Sin embargo, tras realizar pruebas, se ha visto una forma más eficiente de hacerlo.

Este enfoque usa lo que se conoce como **noun chunks**. Este tipo de objeto es una lista formada por los sustantivos y todas sus respectivas palabras que lo describen [22]. Se puede ver el enfoque en el siguiente ejemplo:

> —Sí, claro —respondió Bruno. Siempre había muchas visitas en casa de hombres con uniformes y mujeres con máquinas de escribir.
>
> Sujeto: Bruno
> Noun chunks: [Bruno, muchas visitas, casa, hombres, uniformes, mujeres, máquinas]

De entre todos los sustantivos de la lista de chunks, debe estar el interlocutor. Para identificarlo de entre estos sustantivos, hay que corroborar primero que semánticamente el sustantivo sea un sujeto y posteriormente corroborar si el padre de este sustantivo es el verbo raíz del inciso. spaCy, también permite corroborar árbol de análisis, el cual muestra los hijos o padre de cada palabra. En el caso anterior, el nombre Bruno es la rama hija de respondió e inversamente, respondió es la rama padre de Bruno. Con esto, se puede sacar el interlocutor del diálogo.

Tras obtener quien realiza la acción de hablar, queda solucionar el segundo paso. Para ello, como se ha dicho anteriormente, es necesario reconstruir la oración. Como anteriormente se ha dividido la oración se tienen el parlamento y el inciso separado, por lo que hay que modificar únicamente la parte del inciso. Primeramente, se extrae el verbo del inciso y se pone al final del mismo, para tener el verbo inmediatamente antes del diálogo. Después, se extraen todas las

palabras incluidas en el chunk perteneciente al sujeto de la conversación y se colocan al principio del inciso. Por último, al inciso se le añaden dos puntos, un salto de linea y un tabulado, y a continuación el parlamento. En caso de que el inciso sea largo, se escogerá hasta el primer punto y el resto se colocará posterior al parlamento. Se tiene por lo tanto la siguiente estructura:

Chunk sujeto + inciso + verbo + : + salto linea + tabular + parlamento + [resto inciso]

Su Madre con tristeza nuevamente contestó:
De momento tenemos que cerrarla

3.2.3. Diálogos sin incisos

Este caso es el que contiene la mayor parte de la dificultad, ya que no existe un algoritmo que realice esta tarea con un porcentaje de éxito aceptable. Incluimos en este caso todo diálogo que no tenga inciso, o bien en su inciso contenga lo que se conoce como *anáfora-cero*, que significa que el inciso no contiene ninguna anáfora que haga referencia al orador. Algunos ejemplos son los siguientes:

–¿Qué? –preguntó fingiendo no saber a qué se refería.

–Mira, un bosque –dijo sin hacerle caso.

–Ya lo sé. Pero no podemos hacer nada.

–Te refieres a Padre –expresé.

3.2.3.1. Enfoque

Existen varios modelos distintos que han intentado arreglar este problema de *anáfora-cero* ya que se trata de uno de los mayores problemas a la hora de reorganizar un texto. Un ejemplo interesante de resolución del problema de las anáforas es usar el modelo **GUITAR** [23]. Este modelo esta presente en varios proyectos distintos debido a sus resultados aceptables en el ámbito de la PLN. Aun así, este modelo no se puede usar en el caso actual, la principal razón es que es un modelo enteramente en ingles y no serviría para el español.

El ingles es el idioma predilecto para realizar todos los modelos a día de hoy. Aun así existen modelos multilenguajes o de algún idioma de forma más especifica. Para el caso actual, el español, a falta de un modelo en español se puede buscar de un idioma parecido. Existe un proyecto en italiano llamado **ERNESTA** que busca simplificar frases para los niños. Se trata de un tema parecido al actual y en él se encuentran con el mismo problema, *anáfora-cero*. Para solventar este problema lo que hacen es extraer todos los nombres personales y sus correspondientes verbos, excluyendo los verbos que no pueden ser acción de nadie

como llover. Después se analizan los nombres propios que podrían encajar en los verbos cogiendo hasta tres oraciones anteriores [24].

Antes de aplicar este enfoque al caso, se puede realizar un paso intermedio para conseguir los oradores en primera persona. Estos incisos tienen como sujeto 'yo' en caso de que sea en singular o nosotros en caso de que sea en plural. Esta es la mejor opción posible en vez de insertar el nombre del protagonista, ya que puede conllevar un trabajo extra al necesitar convertir todas las formas verbales de primera a tercera persona. No solo esto, sino que también puede confundir al lector al dar la impresión de que el protagonista no es verdaderamente el protagonista. Esto se puede apreciar en la figura 3.1, en la sección en rojo del mismo.

Para la resolución del caso se van a usar dos enfoques distintos. Una primera que aprovecha las ideas de **ERNESTA** y una segunda que pretende usar *machine learning*. Es interesante probar ambos modelos para ver cómo de fiables son para este caso en particular. En ambos casos se seleccionarán tres oraciones previas al diálogo impersonal.

El primer enfoque consiste en realizar una búsqueda de los nombres previos al diálogo y escoger los últimos que aparezcan. Después se elegirá el nombre personal que más se adecue como sujeto del diálogo. Esto se realizaría únicamente con los diálogos con sujeto omitido. La contrapartida que se ve a simple vista es que no pueden obtener los interlocutores que sean sustantivos más comunes como puede ser *el abuelo* o *la mujer*. Tampoco sería posible que el ultimo nombre que se mencione no sea el personaje que hable. A pesar de esto, merece la pena corroborar como de eficiente puede llegar a ser.

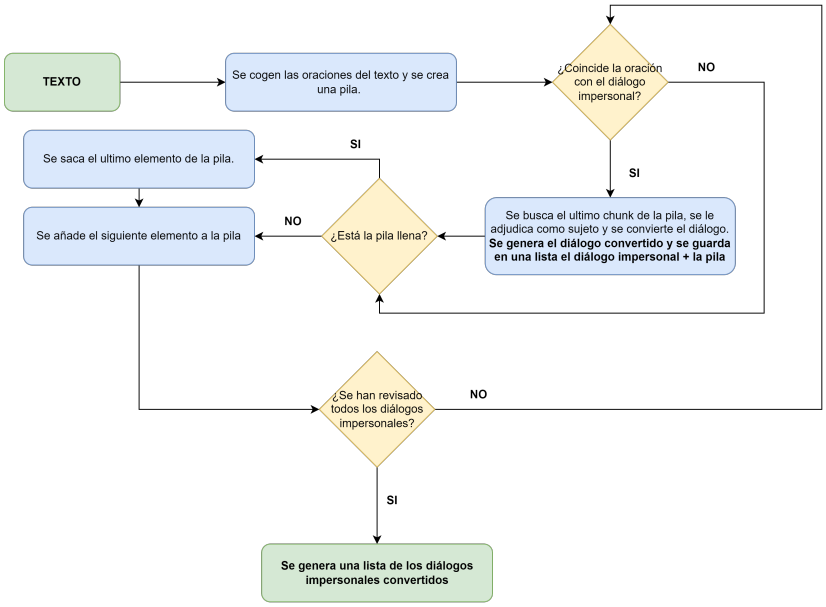

Figura 3.2: *Diagrama de flujo del método 1 del caso 2*

En la figura 3.2 se aprecia de forma gráfica el primer enfoque para este segundo

caso. A simple vista es más simple que el primer caso pero se ha simplificado, debido a que reusa partes del primer caso. Principalmente lo que se reutiliza es la reconstrucción del diálogo a su forma de lectura fácil.

El segundo enfoque consiste en usar un modelo entrenado de *deep learning* para encontrar el sujeto omitido. Para ello se pretende usar el modelo más famoso hoy en día, el modelo *gpt-3.5-turbo*, más conocido como **ChatGPT**. Se ha pensado en este modelo, porque se ha comprobado que en general suele dar buenos resultados y está ya entrenado, por lo que no sería necesario entrenarlo. Como se ha expuesto anteriormente, para que este modelo pueda dar buenos resultados, será necesario proporcionarle el contexto del diálogo, siendo este contexto las tres oraciones previas al diálogo con sujeto omitido. Al igual que el anterior método, no se sabe con exactitud su porcentaje de fallos, por lo que se pretende investigar la cantidad de veces que devuelve correctamente el sujeto.

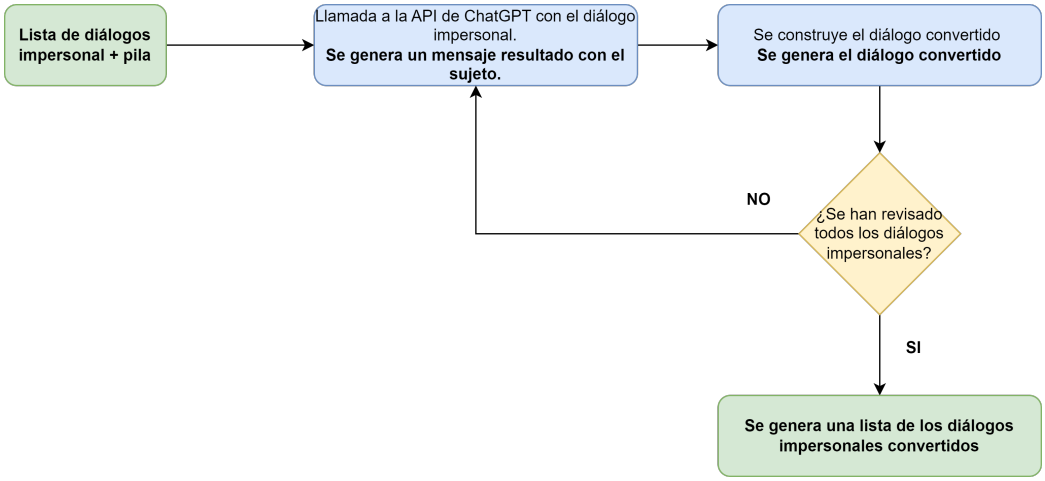

Figura 3.3: *Diagrama de flujo del método 2 del caso 2*

En la figura 3.3 se aprecia de forma gráfica el segundo enfoque para este segundo caso. Al igual que el anterior caso, se ha simplificado para que no resulta tan complejo de entender. Al usar un modelo externo como puede ser el modelo gpt-3.5, no se explicará el proceso interno del mismo.

3.2.3.2. Implementación final

Para las implementaciones del segundo caso, se han usado como base las del primer caso, por lo tanto primero se analizan los diálogos con sujeto y posteriormente los que tienen sujeto omitido.

Para el primer caso de sujetos omitidos, diálogos con incisos en primera persona, se pretende abordar a la vez que transcurre el caso de diálogos con sujetos. El enfoque que se pretende seguir es simple pero efectivo. En el bucle donde se obtiene el verbo raíz del diálogo, se añaden dos funciones extras cuando se encuentre el verbo, ya que se va a comprobar las formas verbales del verbo raíz. Estas funciones comprueban la persona y el numero del verbo. En caso de que el verbo esté en primera persona del singular, el interlocutor será “yo”, en caso de

que esté en primera persona del plural el interlocutor será “nosotros”. Después se continúa con el proceso explicado en el caso de diálogo con inciso, donde se reconstruye el inciso. Para comprobar las formas verbales, spaCy proporciona una herramienta llamada *morph*, la cual analiza morfológicamente una palabra. Tiene muchos atributos como la persona, el numero, el modo y el tiempo entre otros.

Para el siguiente caso, como se ha explicado hay dos métodos. Para ambos métodos, se requiere que previamente se guarde en una lista el conjunto de diálogos que se hayan detectado como sujetos omitidos.

Para el primer método, se va a recorrer nuevamente las oraciones del texto original, pero con la diferencia de que esta vez se va a usar una pila con tres oraciones. Estas oraciones en la pila van rotando a medida que se van analizando las nuevas oraciones hasta que se encuentre uno de los diálogos con sujeto omitido. Cuando se dé este caso, se llamará a una función auxiliar encargada de convertir el diálogo en su versión de lectura fácil. Para ello, usa el mismo algoritmo que se ha estado usando en el caso original, pero, como se tienen las oraciones previas al diálogo, se recorren en busca del ultimo nombre que aparezca. Tras esto, se da por hecho que este nombre es el interlocutor del diálogo donde no existe inciso y se realiza el algoritmo igual que antes. Este proceso se repite con todos los diálogos impersonales existentes en el texto dado.

Para el último modelo se va a usar el modelo de *deep learning* **gpt-3.5-turbo**. Para poder acceder al mismo, es necesario usar la API de la compañía de investigación **OpenAI** [25]. Se ha trabajado con una versión gratuita del chatbot de OpenAI al no contar con los recursos para el uso de una versión superior. Esta versión hubiera reflejado mejores resultados en el experimento. Esta versión gratuita es la llamada **youchat** [26], la cual cuenta con un modelo muy parecido al OpenAI y permite hacer pruebas de forma ilimitada. OpenAI proporciona muchos servicios y funcionalidades distintas, pero el más útil, y el único apropiado para ser usado con el modelo gpt-3.5-turbo es la funcionalidad de *chat*. Esta funcionalidad se basa en lo que se conoce como un chatbot convencional pero con la peculiaridad de que existe un espacio, llamado *system* que sirve para dar ordenes a la inteligencia artificial. Por ejemplo, en el caso a estudiar se pretende que dado un conjunto de oraciones, devuelva el sujeto del último diálogo (el cuál es el diálogo con sujeto omitido). Dado este caso, se le puede ordenar al sistema que realice esta acción para cualquier texto que escriba el usuario.

> **System**: Dime el interlocutor del ultimo diálogo del user en 1 palabra
> **User**: Se asomó y vio el coche en el que habían llegado, así como tres o cuatro coches más de los soldados de Padre, algunos de los cuales andaban por allí, fumando cigarrillos y riendo de algo mientras miraban nerviosos hacia el edificio.
> Un poco más allá estaba el camino de la casa, y más allá había un bosque que parecía ideal para explorar.
> –Bruno, ¿quieres hacer el favor de explicarme qué has querido decir con ese último comentario? –preguntó Gretel.
> –Mira, un bosque –dijo sin hacerle caso. <– *Diálogo con sujeto omitido*

Assistant: Bruno <– *Respuesta de la Inteligencia Artificial*

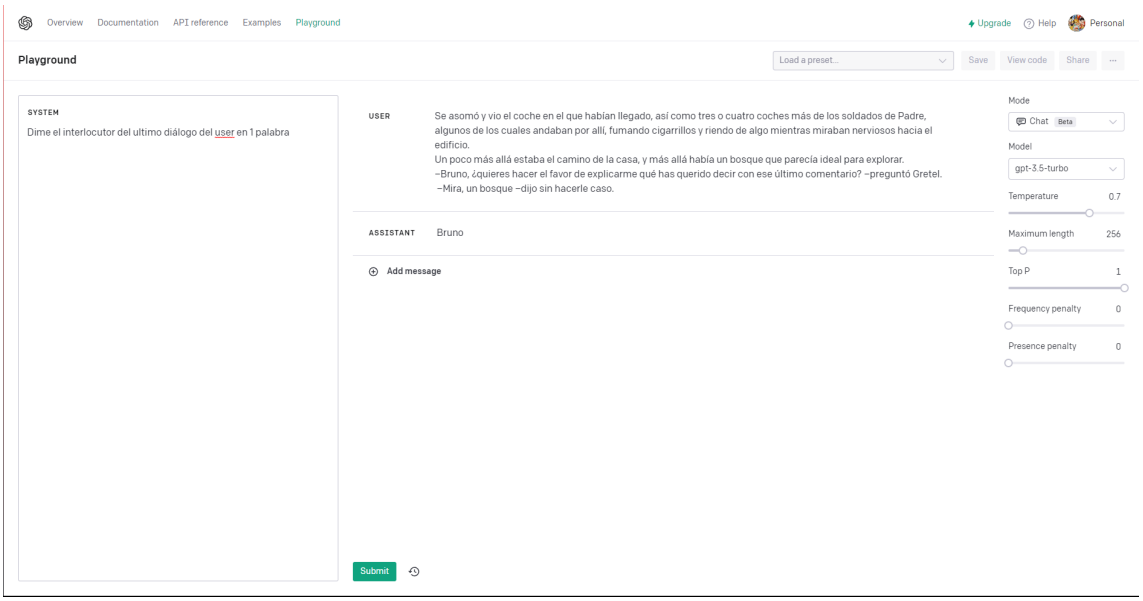

Figura 3.4: *Playground de OpenAI*

En la figura 3.4 se puede apreciar la aplicación web de OpenAI que ofrece los servicios de las APIs.

Tras realizar la llamada a la API, se obtiene un resultado, el cual será el orador que se busca. A partir de aquí el proceso es el mismo, hay que reordenar la frase y colocar este nuevo orador al inicio.

3.3. Casos de prueba

Como en cualquier proceso de investigación, es necesario realizar pruebas de todos los casos imaginables y surgen posibles cambios a raíz de cambios necesarios. Es por este motivo por el que una de las secciones más importantes son las pruebas, aquí se puede comprobar la evolución de la investigación y entender correctamente las decisiones tomadas.

3.3.1. Modelos

El primer problema al que se enfrenta cualquier proyecto de PLN de estas magnitudes es encontrar el modelo que más se adecue. Para ello, es necesario realizar unos descartes iniciales y realizar pruebas a lo largo de la investigación. Como se ha mencionado en apartados anteriores, tras una fase de descartes inicial, se centró la atención en dos modelos, el modelo mediano y el modelo grande de español de spaCy, los cuales se han probado para determinar cual es más beneficioso usar en este caso [27].

La primera prueba se ha realizado en el caso uno, en la tarea de buscar el sujeto. En ese momento se estaban realizando pruebas con un corpus mucho más

grande y complicado. Posteriormente se cambió a un corpus más sencillo pensando en los micro relatos. La principal razón para ello es no abarcar demasiado, ya que puede distorsionar el objetivo del proyecto. Volviendo a los modelos, se realizó una prueba que mostrase todos los sujetos de este corpus para ambos modelos y se comparó. En oraciones simples los dos modelos actúan de la misma forma, donde difieren es en oraciones mas complejas. Un ejemplo claro es el siguiente:

> —Haz el favor de no coartar el pobre Nicolás —respondió a la señora el señor Savolta, y dirigiéndose al señor Claudedeu

En este ejemplo el sujeto debe ser el señor Savolta, ya que es quien realiza la acción de responder. El modelo medio, interpreta que el señor Savolta es un complemento directo de la señora. Por otro lado, el modelo grande lo interpreta correctamente. Como resumen de las pruebas, en frases complejas, el modelo grande tiene menos errores, destacando cuando aparece un nombre propio. Por otro lado, el medio consigue encontrar mejor los sujetos que son sustantivos corrientes. De las 1430 casos que se han probado, han diferido en el 4,75% de las veces.

Con respecto a los chunks, ambos modelos los interpretan exactamente igual. Se han realizado una serie de pruebas con las versiones finales del código y los resultados son los mismos.

Por lo tanto, se puede concluir que ambos modelos se comportan de la misma forma para el algoritmo final, pero, dado que en casos más complejos rinde mejor el modelo grande se usará de cara a futuras mejoras del algoritmos.

3.3.2. Caso 1: Diálogo con inciso

Se han realizado dos tipos de pruebas distintas, una para probar la búsqueda de sujeto y otra para la creación de la oración final. Esto es debido a que cada parte tiene suficiente contenido y capacidad de error como para necesitar dividirlo.

Se han realizado varias pruebas distintas, la más destacable es la conversión de todos los diálogos del libro *El niño con el Pijama de Rayas* en su versión de Lectura Fácil, dado que los diálogos no están convertidos. Se van a englobar todos los casos de pruebas alrededor de este libro, aunque cada parte tiene sus pruebas especificas. Por último, se ha obviado el caso de identificar el diálogo en el texto, ya que pertenecen a la sección previa al algoritmo.

3.3.2.1. Búsqueda del sujeto

El primer caso de pruebas que se ha decidido realizar es convertir los diálogos del capitulo uno del libro. En este primer capítulo se encuentran los siguientes diálogos:

> 1º –¿Qué haces? –preguntó Bruno a la criada con mucha educación, aunque no le gustaba que le tocaran sus cosas. Su Madre le decía que tratara con respeto a la criada; todo lo contrario de lo que hacía su

Padre.
2º –Madre, ¿por qué María está revolviendo mis cosas?–preguntó Bruno.
3º –Está haciendo las maletas –respondió la Madre.
4º –¿Por qué? ¿Qué he hecho?
5º –Madre, ¿qué pasa? ¿Vamos a mudarnos?
6º –Ven conmigo al comedor, allí hablaremos –contestó la Madre.
7º –Mira, Bruno, no te preocupes, vas a vivir una gran aventura –le explicó su Madre.
8º –¿Qué aventura? ¿Vais a mandarme a algún sitio? –preguntó Bruno a su Madre.
9º –No. No te vas tú sólo. Nos vamos todos. Tú, Gretel, tu padre y yo –le respondió su Madre.
10º –¿Adónde nos vamos? ¿Por qué no nos quedamos aquí?
11º –Es por el trabajo de tu Padre, ya sabes que es muy importante, ¿verdad? –le aclaró su Madre.
12º –Sí, claro –respondió Bruno. Siempre había muchas visitas en casa de hombres con uniformes y mujeres con máquinas de escribir.
13º –A veces, cuando alguien es muy importante, su jefe le pide que vaya a algún sitio para hacer un trabajo muy especial–siguió comentando la Madre.
14º –¿Qué clase de trabajo? –quiso saber Bruno, ya que no sabía realmente en qué trabajaba su Padre.
15º –Es un trabajo muy importante, un trabajo para un hombre muy especial. Lo entiendes, ¿verdad? –le contestó su Madre.
16º –¿Tenemos que ir todos? –preguntó Bruno.
17º –Sí claro, no querrás que Padre se vaya solo y esté triste–añadió su Madre con voz muy seria.
18º –No, claro que no –negó Bruno con rotundidad.
19º –Padre sentiría nuestra ausencia si no vamos con él–añadió la Madre.
20º –Pero, ¿y la casa? ¿Quién cuidará de la casa mientras estemos fuera? –siguió preguntando Bruno.
21º –De momento tenemos que cerrarla –contestó con tristeza nuevamente su Madre.
22º –¿Y está muy lejos ese sitio al que tenemos que ir? –volvió a preguntar Bruno.
23º –Sí, Bruno, está muy lejos. Nos vamos fuera de Berlín –aclaró su Madre.
24º –¿Y la escuela? ¿Qué pasa con mis 3 mejores amigos: Karl, Martín y Daniel? –preguntó preocupado Bruno.
25º –Tendrás que despedirte de tus amigos durante un tiempo. Pero volverás a verlos dentro de poco. Ya harás nuevos amigos en el lugar al que vamos –dijo la Madre a Bruno.
26º –¿Despedirme de ellos? ¡Pero si son mis 3 mejores amigos! –protestó Bruno enfadado y con un tono de voz alto.

Para estos casos se obtienen correctamente los sujetos en una lista siendo los

siguientes:

Sujeto1: Bruno
Sujeto2: Bruno
Sujeto3: La Madre
Sujeto4: Sujeto Omitido
Sujeto5: Sujeto Omitido
Sujeto6: La Madre
Sujeto7: Su Madre
Sujeto8: Bruno
Sujeto9: Su Madre
Sujeto10: Sujeto Omitido
Sujeto11: Su Madre
Sujeto12: Bruno
Sujeto13: Sujeto Omitido
Sujeto14: Bruno
Sujeto15: Su Madre
Sujeto16: Bruno
Sujeto17: Su Madre
Sujeto18: Bruno
Sujeto19: La Madre
Sujeto20: Bruno
Sujeto21: Su Madre
Sujeto22: Bruno
Sujeto23: Su Madre
Sujeto24: Bruno
Sujeto25: La Madre
Sujeto26: Bruno

El 100 % de los sujetos se corresponden con las oraciones con el algoritmo empleado. A pesar de esto, se han seguido haciendo pruebas y existen casos más complejos donde no funciona como se desea. Estos casos suelen alejarse un poco de la estructura de diálogo sencilla de los microcuentos, aun así, es interesante comprobar como funciona en estos casos para un posible futuro uso.

Un ejemplo de estos casos es el siguiente:

> —No faltaría más, señor —respondió el viejo con la condescendencia del que, no habiendo cedido en lo más, se complace en ceder en lo menos—, aquí tiene.

Este caso se detectaría con el algoritmo actual como un diálogo con sujeto omitido, y por lo tanto iría al segundo caso. La principal razón para esto es debido a que el sujeto del inciso debería ser *el viejo*, pero el modelo detecta viejo en este caso como un adjetivo en vez de como un sustantivo. Con respecto a los chunks del inciso son: *la condescendencia* y *que*. Estos chunks no son posibles sujetos del inciso ya que no tienen como raíz al verbo responder.

Este error suele darse comúnmente cuando el sujeto en cuestión puede actuar

a su vez de adjetivo u otra forma. Otro error que puede darse es la aparición de perífrasis verbal. Un ejemplo del mismo es el siguiente diálogo que se encuentra en el cuarto capitulo del libro:

> —¿Y dónde están las niñas? ¿Y las madres? ¿Y las abuelas? —siguió preguntando Gretel.

En este caso el sujeto claramente es Gretel, pero al tener una forma verbal compuesta como puede ser *seguir preguntando* el modelo detecta como sujeto raíz el verbo seguir. Esto es correcto sintácticamente, el problema viene al ver cual es la cabeza a la que apunta el nombre Gretel. En vez de apuntar al verbo raíz, apunta a preguntando que actúa como un complemento clausal de seguir.

3.3.2.2. Estructuración final

Con los diálogos pertenecientes al capitulo uno del libro se obtienen las sigueintes conversiones:

> 1º Bruno a la criada con mucha educación, aunque no le gustaba que le tocaran sus cosas preguntó:
> ¿Qué haces?
> Su Madre le decía que tratara con respeto a la criada; todo lo contrario de lo que hacía su Padre
>
> 2º Bruno preguntó:
> Madre, ¿por qué María está revolviendo mis cosas?
>
> 3º La Madre respondió:
> Está haciendo las maletas
>
> 4º Sujeto Omitido
>
> 5º Sujeto Omitido
>
> 6º La Madre contestó:
> Ven conmigo al comedor, allí hablaremos
>
> 7º Su Madre le explicó:
> Mira, Bruno, no te preocupes, vas a vivir una gran aventura
>
> 8º Bruno a su Madre preguntó:
> ¿Qué aventura? ¿Vais a mandarme a algún sitio?
>
> 9º Su Madre le respondió:
> No. No te vas tú sólo. Nos vamos todos. Tú, Gretel, tu padre y yo
>
> 10º Sujeto Omitido
>
> 11º Su Madre le aclaró:
> Es por el trabajo de tu Padre, ya sabes que es muy importante, ¿verdad?

12º Bruno respondió:
Sí, claro
Siempre había muchas visitas en casa de hombres con uniformes y mujeres con máquinas de escribir

13º Sujeto Omitido

14º Bruno ya que no sabía realmente en qué trabajaba su Padre quiso saber:
¿Qué clase de trabajo?

15º Su Madre le contestó:
Es un trabajo muy importante, un trabajo para un hombre muy especial. Lo entiendes, ¿verdad?

16º Bruno preguntó:
¿Tenemos que ir todos?

17º Su Madre con voz muy seria añadió:
Sí claro, no querrás que Padre se vaya solo y esté triste

18º Bruno con rotundidad negó:
No, claro que no

19º La Madre añadió:
Padre sentiría nuestra ausencia si no vamos con él

20º Bruno siguió preguntando:
Pero, ¿y la casa? ¿Quién cuidará de la casa mientras estemos fuera?

21º Su Madre con tristeza nuevamente contestó:
De momento tenemos que cerrarla

22º Bruno a volvió preguntar:
¿Y está muy lejos ese sitio al que tenemos que ir?

23º Su Madre aclaró:
Sí, Bruno, está muy lejos. Nos vamos fuera de Berlín

24º Bruno preocupado preguntó:
¿Y la escuela? ¿Qué pasa con mis 3 mejores amigos: Karl, Martín y Daniel?

25º La Madre a Bruno dijo:
Tendrás que despedirte de tus amigos durante un tiempo. Pero volverás a verlos dentro de poco. Ya harás nuevos amigos en el lugar al que vamos

26º Bruno enfadado y con un tono de voz alto protestó:
¿Despedirme de ellos? ¡Pero si son mis 3 mejores amigos!

En el caso de reconstrucción, al no existir unas reglas fijas y universales que seguir para que un diálogo este en formato de lectura fácil, se ha optado por este.

El formato se basa en dejar el parlamento intacto para su posterior transformación en caso de ser necesario. Para el caso del inciso, poner al inicio el sujeto y al final el verbo, además en caso de existir más inciso o que el inciso sea muy largo, se divide y se pondrá el resto posteriormente al parlamento. Un ejemplo de esto es el primer diálogo.

Este método no tiene casos de errores como tal, ya que el problema principal es encontrar la forma donde todos los incisos nuevos sean claros. En el decimoctavo diálogo, se puede ver como el inciso muestra la información de forma que se entiende, pero puede que sea necesario añadir algún determinante extra.

También existen casos donde las perífrasis verbales pueden dar problemas como el caso del diálogo vigesimosegundo. En este caso, el algoritmo no ha detectado correctamente la existencia de estos perífrasis verbales.

En el libro hay un total de 663 diálogos, de los cuales 84 de ellos han sido detectados como *Sujeto Omitido*, los cuales se analizarán en las siguientes pruebas. De estos diálogos, el 4,07 % de ellos se han convertido a lectura fácil con algún fallo. Muchos de estos fallos son debidos al modelo y muchos otros son debidos a la presencia de perífrasis verbales.

Los errores encontrados en los 663 diálogos analizados se pueden englobar en las siguientes categorías:

- Falso Sujeto Omitido
- Perífrasis Verbales
- Mala estructuración

Estas categoría se pueden apreciar en la figura 3.5. En esta gráfica se puede apreciar la cantidad de fallos existentes para cada categoría.

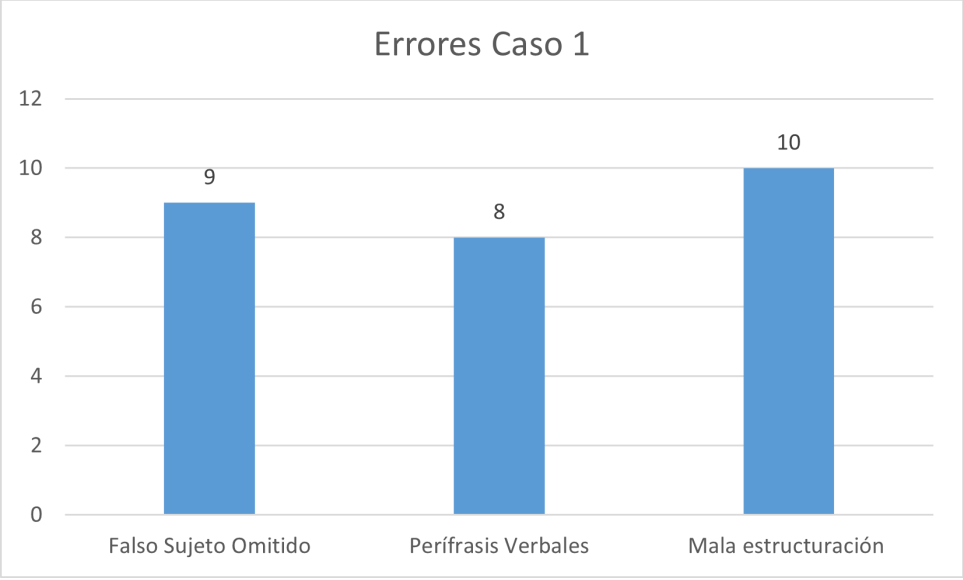

Figura 3.5: *Errores caso 1*

3.3.3. Caso 2: Diálogo sin inciso

Para este caso, se han probado los tres enfoques expuestos. Nuevamente se ha usado el libro de *El niño con el Pijama de Rayas* en su versión de Lectura Fácil. De los 84 diálogos del libro que entran en esta categoría, 8 de ellos son erratas debidas a fallos con el modelo al tener perífrasis verbales. Aun así, es interesante incluirlos en los métodos para ver cómo reaccionan.

3.3.3.1. Diálogos en primera persona

Este caso es sencillo y no requiere de muchas pruebas. Debido a que en el libro no existe ningún diálogo cuyo inciso esté en primera persona, se ha recurrido a un conjunto de relatos cortos que también se usan como corpus, además de otros cuentos infantiles. El caso que se va a tratar es el relato titulado *Mi pierna derecha* de Juan José Millás [28].

En este relato encontramos el siguiente diálogo:

> –¿Te has quedado sin gasolina? –pregunté.

El orador de este diálogo es el propio narrador, es por ello que esta en primera persona del singular. La conversión da por lo tanto:

> Yo pregunté:
> ¿Te has quedado sin gasolina?

En general estos casos los detecta bien, ya que depende del análisis morfológico del modelo. Este análisis suele ser preciso y correcto.

3.3.3.2. Método 1: Búsqueda de nombres

Se van a usar como casos de prueba todos los diálogos impersonales del libro. Una muestra de ellos son:

> 1º –¿Por qué? ¿Qué he hecho?
> 2º –Madre, ¿qué pasa? ¿Vamos a mudarnos?
> 3º –¿Adónde nos vamos? ¿Por qué no nos quedamos aquí?
> 4º –A veces, cuando alguien es muy importante, su jefe le pide que vaya a algún sitio para hacer un trabajo muy especial–siguió comentando la Madre.
> 5º –A nosotros no nos corresponde pensar. Ciertas personas deciden por nosotros.
> 6º –Me parece que nos hemos equivocado. Creo que lo mejor será olvidar todo esto y volver a casa.
> 7º –¿Por qué no subes y ayudas a María a deshacer las maletas?
> 8º –Si quieres separa toda esa ropa y colócala en la cómoda
> 9º –le contestó María señalando una bolsa que contenía todos sus calcetines, camisetas y calzoncillos.
> 10º –No es muy bonito, ¿verdad?
> 11º –Bueno, sí. Ahora puede parecer horrible. Pero cuando arreglemos

un poco la casa seguro que no nos parecerá tan mal. Le oí decir a Padre que quienes vivían aquí en Auchviz antes que nosotros perdieron su empleo muy deprisa y no tuvieron tiempo de arreglar la casa para nosotros.
12º –Es el nombre de la casa: Auchviz.
13º –¿Qué has dicho?
14º –Mira, un bosque –dijo sin hacerle caso.
15º –¿Qué? –preguntó fingiendo no saber a qué se refería.
16º –¿Y dónde están las niñas? ¿Y las madres? ¿Y las abuelas? –siguió preguntando Gretel.

De estos diálogos, algunos son errores acarreados por pasos anteriores como el decimosexto diálogo. Las correspondientes conversiones de estos diálogos son los siguientes:

1º Bruno dijo:
¿Por qué? ¿Qué he hecho?

2º Bruno dijo:
Madre, ¿qué pasa? ¿Vamos a mudarnos?

3º Su Madre dijo:
¿Adónde nos vamos? ¿Por qué no nos quedamos aquí?

4º Bruno la Madre siguió comentando:
A veces, cuando alguien es muy importante, su jefe le pide que vaya a algún sitio para hacer un trabajo muy especial

5º La Madre de Bruno dijo:
A nosotros no nos corresponde pensar. Ciertas personas deciden por nosotros.

6º Bruno dijo:
Me parece que nos hemos equivocado. Creo que lo mejor será olvidar todo esto y volver a casa.

7º Madre dijo:
¿Por qué no subes y ayudas a María a deshacer las maletas?

8º María dijo:
Si quieres separa toda esa ropa y colócala en la cómoda

9º María dijo:
le contestó María señalando una bolsa que contenía todos sus calcetines, camisetas y calzoncillos.

10º Gretel dijo:
No es muy bonito, ¿verdad?

11º Bruno dijo:
Bueno, sí. Ahora puede parecer horrible. Pero cuando arre-

glemos un poco la casa seguro que no nos parecerá tan mal. Le oí decir a Padre que quienes vivían aquí en Auchviz antes que nosotros perdieron su empleo muy deprisa y no tuvieron tiempo de arreglar la casa para nosotros.

12º Bruno dijo:
Es el nombre de la casa: Auchviz.

13º Gretel dijo:
¿Qué has dicho?

14º Gretel sin caso hacerle dijo:
Mira, un bosque

15º Bruno no saber a qué se refería preguntó fingiendo:
¿Qué?

16º Bruno Gretel siguió preguntando:
¿Y dónde están las niñas? ¿Y las madres? ¿Y las abuelas?

De estos dieciséis casos seis de ellos son incorrectos entre ellos los diálogos: 3, 4, 11, 12, 15 y 16. Estos errores engloban problemas anteriores (16) como problemas en la composición del inciso (15) y, lógicamente, errores en el sujeto (3).

De los 84 diálogos, 52 de ellos se han convertido de forma errónea. Esto indica que en este libro el porcentaje de error ha sido del 61,90 %.

Estos 52 fallos son principalmente fallos del propio método cogiendo un sujeto erróneo, pero también se incluyen los 9 errores del caso 1 que acarrea.

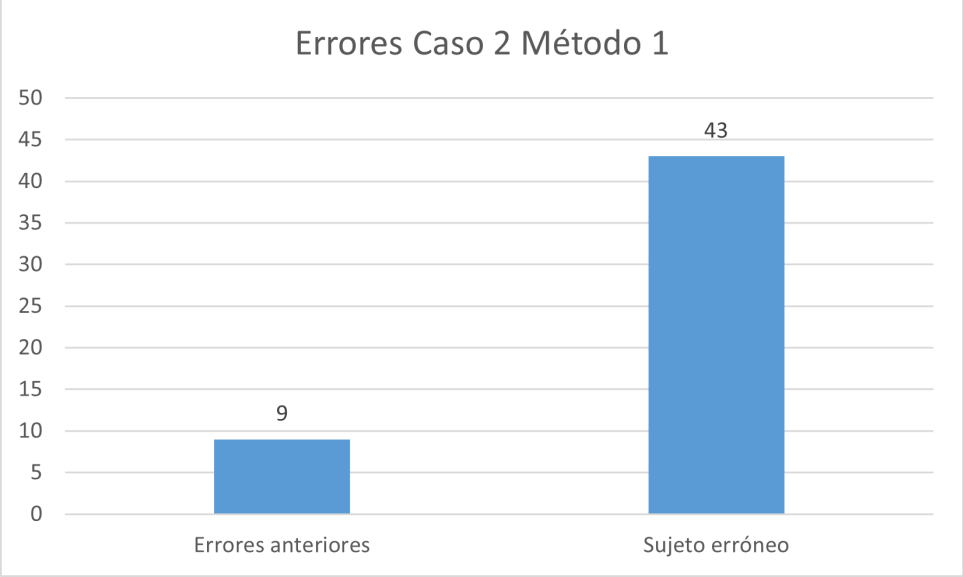

Figura 3.6: *Errores Caso 2 Método 1*

En la figura 3.6 se pueden ver que tipos de fallos son cada uno de los 52 errores. Los 9 errores anteriores son errores acarreados del caso anterior por lo que son menos graves en esta sección.

3.3.3.3. Método 2: Uso de modelo de deep learning

Para comparar este segundo método con el método anterior se van a usar las dieciséis muestras anteriores. Al tratarse de una API de un modelo de *deep learning*, no se pueden realizar muchas llamadas consecutivas sin un importe económico.

A diferencia del anterior método que de los dieciséis casos tenia seis errores, este método tiene tres errores. Uno de estos errores, al igual que el anterior, es debido a un fallo en el modelo PLN cuando existe un perífrasis verbal (caso 16).

De los 84 diálogos analizados, tan solo 20 de ellos se han convertido de forma errónea. Esto indica que en este libro el porcentaje de error ha sido del 23,81 %, datos muy positivos en comparación con el anterior método. De los 20 errores, solamente 6 de ellos son acarreados de errores anteriores. Esto indica que ha solucionado 3 errores anteriores y puede potencialmente solucionar errores parecidos.

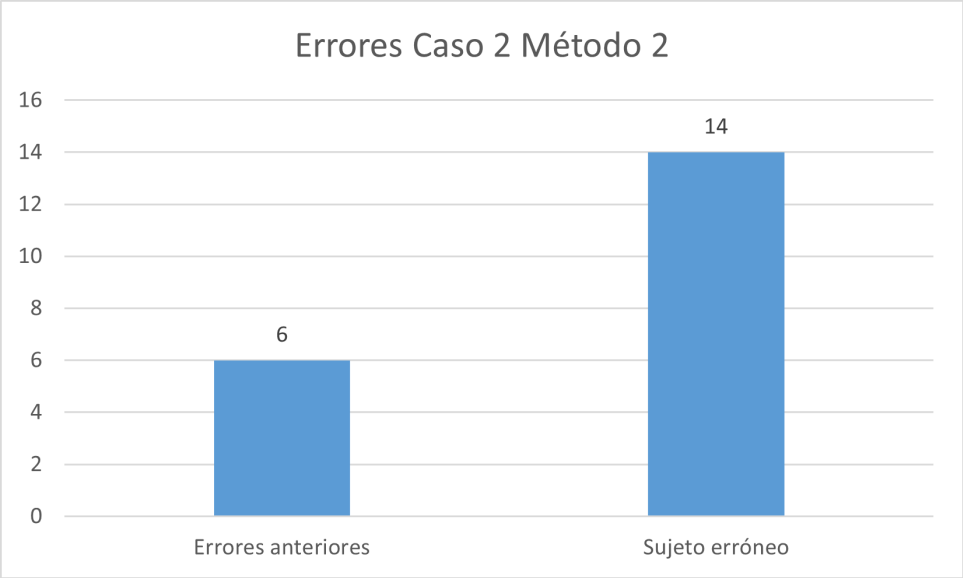

Figura 3.7: *Errores Caso 2 Método 2*

Al igual que en el apartado 3.3.3.2, se puede observar en la figura 3.7 los tipos de errores existentes en este método. Se puede apreciar gráficamente la disminución de los errores acarreados del caso anterior mediante el uso de este segundo método.

3.4. Resultados

Para finalizar, en la próxima gráfica 3.8 se puede apreciar de una forma más visual el porcentaje de fallo que ha tenido cada método para el libro usado.

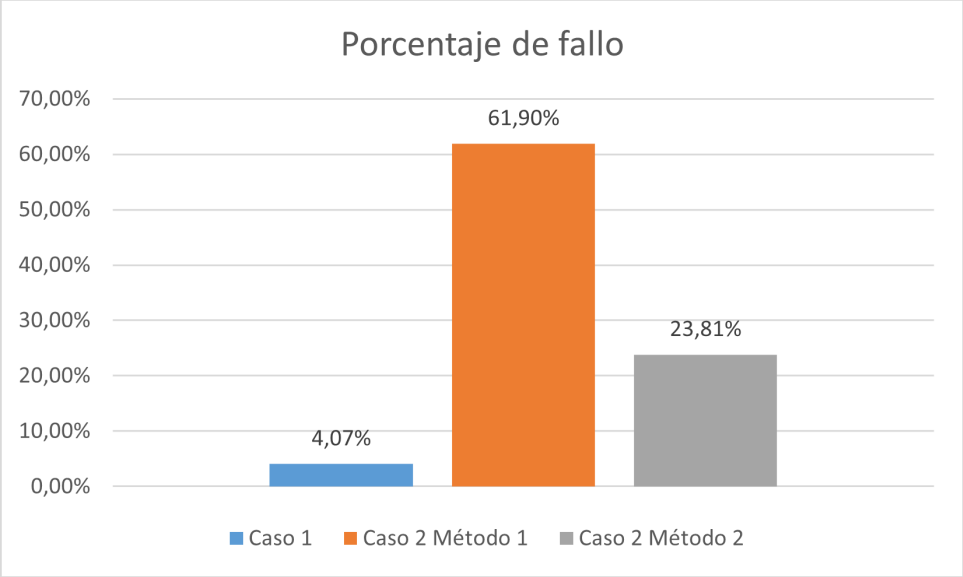

Figura 3.8: *Comparación de errores de métodos*

Como se ha explicada en los apartados 3.3.2 y 3.3.3, el primer caso es donde menos porcentaje de fallos se ha conseguido y el segundo caso donde más porcentaje se ha obtenido. De este segundo caso, se distingue el primer método con más del 60 % de fallos con el segundo método que tiene mejores resultados, sin llegar a superar el 25 % de fallos.

Para el primer caso es necesario realizar un análisis de los parámetros predictivos, siendo para este caso siguientes parámetros:

- **Verdadero Positivo** (VP o TP en inglés): 579 casos
- **Falso Positivo** (FP): 0 casos
- **Falso Negativo** (FN): 9 casos
- **Verdadero Negativo** (VN o TN en inglés): 75 casos

Estos parámetros tienen un significado concreto. El verdadero positivo incluye los ejemplos donde se detecta correctamente que el diálogo tiene un sujeto; el falso positivo incluye los ejemplos donde se detecta que el diálogo tiene un sujeto cuando en realidad es sujeto omitido; falso negativo incluye los ejemplos donde se detecta que el diálogo tiene un sujeto omitido cuando en realidad tiene sujeto y el verdadero negativo incluye los ejemplos donde detecta correctamente que el diálogo tiene sujeto omitido. Por lo tanto se tiene que la detección de sujeto tiene una **exactitud**[2] del 98.64 %.

[2]Exactitud (Accuracy) = $\frac{VP+VN}{VP+FP+FN+VN}$

Posteriormente a la detección de sujeto se realizan conversiones. Como se ha expuesto en el 3.3.2, de los 663 diálogos, 27 de ellos han sido detectados con alguna serie de error. Por lo tanto, se obtiene que tiene una tasa de éxito[3] del **95.93%**. Cabe destacar que este algoritmo puede mejorar si mejora la exactitud de la detección del sujeto.

Para el segundo caso, expuesto en el apartado 3.3.3, se parten de los sujetos omitidos detectados en el primer caso, siendo un total de 84 diálogos. En este caso existen dos métodos distintos. El primer método, que consiste en usar el ultimo nombre propio anterior al diálogo con sujeto omitido, tiene un total de 52 errores. Esto da como resultado una tasa de éxito del **38.10%**. Por otro lado, el segundo método, que consiste en usar deep learning para obtener el sujeto, tiene un total de 20 errores. Esto da como resultado una tasa de éxito del **76.19%**. Estas tasas de éxito se pueden aumentar con una mejora en la detección del sujeto, ya que 9 de los casos de errores son falsos negativos.

[3]Tasa de éxito = $\frac{Total - Errores}{Total}$

Capítulo 4

Conclusiones

Los resultados que se han encontrado son más esperanzadores de lo esperado inicialmente. Esto es debido a la gran dificultad existente en el mundo del PLN y más aún al haber tan pocos proyectos que involucren un tema tan concreto y a la vez tan generalista como un diálogo.

El primer caso era el más sencillo y por lo tanto se han obtenido unos resultados favorables. Esto no solo indica que este caso se puede investigar más a fondo y llevar a la practica, sino que muestra como el modelo de spaCy usado funciona correctamente. Se destaca especialmente la función **chunk**, la cual ha sido especialmente útil y se le puede sacar más partido de cara a un futuro.

Aún así, existen situaciones en las que puede dar problemas. El caso por excelencia es el de diálogos muy complicados a nivel sintáctico, los cuales pueden confundir al modelo, haciendo que categorice incorrectamente algunos tokens. Otra situación muy repetida en los casos de prueba son los fallos que se producen en algunos casos cuando existe un verbo compuesto. Aquí el modelo puede equivocarse al categorizar a uno de los dos verbos como raíz y al otro como complemento. Esto puede ser un problema incluso en diálogos simples como alguno que se ha visto anteriormente.

En el segundo caso, ambos modelos han sido muy interesantes y han dado buenas conclusiones. Aunque los resultados no sean buenos, en algún caso sigue siendo información útil que se puede usar en un futuro. Al no haber usado un corpus de diálogos impersonales tan extensos, puede que los datos obtenidos estén incompletos. Aún así, sirven como una primera aproximación y es suficiente para poder sacar conclusiones del mismo.

Para el primer modelo los datos han sido bajos. Esto ya se esperaba debido a los resultados presentes en los artículos de algoritmos parecidos. Tener más del 60 % en errores son datos muy pobres, lo cual indica que este método no se puede usar. Puede ser que para casos más específicos o con más restricciones puede bajar este porcentaje.

Por otro lado, el segundo modelo tiene datos más positivos. Se ha reducido un 40 % el porcentaje de error estando en 20 %. Estos datos son suficientes como

para continuar investigando por esta rama. Al estar los modelos de *deep learning* en auge a día de hoy este método tiene más futuro y puede mejorar. Los resultados habrían sido más favorables si se hubiese usado una versión profesional del modelo de OpenAI.

La desventaja de este método es que para usarlo a una escala más elevada es necesario obtener la versión profesional de las APIs de los modelos de *deep learning*. Otro fallo que se ha dado en algunas situaciones es el caso de que existan varios nombres en el texto que se envía al modelo, es posible que coja el nombre equivocado.

Se debe destacar que la parte más complicada de este proyecto es el preprocesado de los datos. Dado que el TFG es continuación de otros TFGs anteriores, la forma de introducir los datos es fija, mediante un .txt o las frases en crudo. En caso de necesitar usar un .pdf, un .doc o un .xlsx para extraer los datos puede complicarse más el asunto. La principal razón para esto es la modificación de los saltos de lineas que crean estos documentos. Como una forma vital para identificar un diálogo es que vaya precedido por un salto de línea, si se modifican puede provocar muchos problemas.

Las pruebas se han corroborado de forma minuciosa por el ojo humano. La principal razón de esto es debido a que las pruebas usadas en anteriores TFGs no sirven. Al tratarse de la misma frase pero con el orden cambiado, el significado se mantendrá igual, por lo que no se puede corroborar de esta forma si la conversión es correcta. Esto provoca que se necesite más tiempo para cada prueba e imposibilita el poder hacer una gran cantidad de ellas por el momento.

Por último, es necesario mencionar la gran utilidad que muestra este tipo de proyectos para la sociedad actual, ya que es una forma de mejorar las medidas de accesibilidad universal. La metodología de Lectura Fácil es capaz de beneficiar a un gran colectivo de personas, en el que se destacan aquellos con algún tipo de discapacidad intelectual y cognitiva. Aun así, existen más grupos que pueden beneficiarse como personas mayores o extranjeros con dificultades para el lenguaje. Esta metodología tiene una estrecha relación a su vez con dos de los Objetivos de Desarrollo Sostenible de la actual Agenda 2030.

Capítulo 5

Trabajos Futuros

En este TFG, al tratarse de un trabajo de investigación, existen numerosos elementos mejorables y por hacer en un futuro.

Primeramente, será conveniente tratar en un futuro datos que no vengan únicamente en un archivo .txt. Para ello, será necesario encontrar alguna forma para mantener los saltos de lineas en los pdfs o docx narrativos. El principal problema es la forma de crear estos pdfs que puede provocar el añadir saltos de lineas extras que pueden entorpecer el trato con los datos. Lo ideal sería crear una función única y exclusivamente para tratar estos saltos de lineas extras que no aportan.

Otro posible trabajo para un futuro es llevar el caso 1 a situaciones más difíciles del trabajo actual. Es decir, se puede investigar su posible implementación no solo en narrativas sencillas como microcuentos, sino en novelas o incluso artículos periodísticos o de otra índole. Esto se puede investigar y comprobar la capacidad de implementación. Para esto, se puede usar el corpus de la RAE original usado en este trabajo, ya que trata con diálogos más complejos que aparecen en novelas.

Como se ha explicado a lo largo del trabajo, han surgido numerosos errores debido al modelo usado de spaCy. Las técnicas PLN están en continuo avance y por lo tanto se están creando nuevos y más poderosos modelos. Muchos de estos modelos son en ingles, pero muchos otros en otros idiomas como el español o una lengua parecida al mismo como el italiano. Por esta razón, al existir nuevos modelos pueden ser usados en este mismo TFG para mejorar los resultados. Incluso en caso de mejorar los resultados enormemente, se pueden llevar a situaciones más complicadas como la expuesta en el anterior párrafo.

Bibliografía

[1] "Making information accessible for all | european blind union," www.euroblind.org, 2022. [Online]. Available: https://www.euroblind.org/publications-and-resources/making-information-accessible-all#Text

[2] G. Muñoz, "Lectura fácil: Métodos de redacción y evaluación," www.plenainclusion.org, 2012. [Online]. Available: https://www.plenainclusion.org/sites/default/files/lectura-facil-metodos.pdf

[3] D. Jurafsky and J. H. Martin, "Speech and language processing," web.stanford.edu, 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

[4] G. G. Chowdhury, "Natural language processing," strathprints.strath.ac.uk, 2003. [Online]. Available: https://strathprints.strath.ac.uk/2611/1/strathprints002611.pdf

[5] "Understanding semantic analysis," www.geeksforgeeks.org, 2021. [Online]. Available: https://www.geeksforgeeks.org/understanding-semantic-analysis-nlp/

[6] "An end to end guide on nlp pipeline," www.analyticsvidhya.com, 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/06/an-end-to-end-guide-on-nlp-pipeline/#:~:text=NLP%20Pipeline%20is%20a%20set,and%20Pipeline%20is%20non-linear.

[7] D. Jurafsky and J. H. Martin, "Naive bayes and sentiments classification," web.stanford.edu, 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/4.pdf

[8] ——, "Regular expressions, text normalization, edit distance," web.stanford.edu, 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/2.pdf

[9] ——, "Sequence labeling for parts of speech and named entities," web.stanford.edu, 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/8.pdf

[10] T. Nasukawa and J. Yi, "Sentiment analysis: capturing favorability using natural language processing," dl.acm.org, 2003. [Online]. Available: https://dl.acm.org/

doi/pdf/10.1145/945645.945658?casa_token=xKzyhxBKX04AAAAA:
GiHubju4QjLLOHWfsUerMlkRtZ8t04xiMocNZCPW-d3a7XMRoYUrxYOzTwt3wqe5rIbzPCtKI

[11] "El principito," planetafacil.plenainclusion.org, 2019. [Online]. Available: https://planetafacil.plenainclusion.org/wp-content/uploads/2019/07/el_principito_lf_1.0.pdf

[12] "Making the web accessible," www.w3.org, 2023. [Online]. Available: https://www.w3.org/WAI/

[13] "Accesibilidad web: Wcag 2.0," accesibilidadweb.dlsi.ua.es, 2023. [Online]. Available: http://accesibilidadweb.dlsi.ua.es/?menu=wcag-2.0

[14] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. LeTraon, "A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate," www.researchgate.net, 2019. [Online]. Available: https://www.researchgate.net/profile/Sylvain-Kubler/publication/337977695_A_Replicable_Comparison_Study_of_NER_Software_StanfordNLP_NLTK_OpenNLP_SpaCy_Gate/links/5f52b5d3299bf13a31a07658/A-Replicable-Comparison-Study-of-NER-Software-StanfordNLP-NLTK-OpenNLP-SpaCy-Ga
pdf

[15] "spacy 101: Everything you need to know," spacy.io, 2023. [Online]. Available: https://spacy.io/usage/spacy-101#features

[16] "Universal dependencies," universaldependencies.org, 2021. [Online]. Available: https://universaldependencies.org

[17] "Cómo escribir un diálogo: La puntuación," www.escueladeescrituracreativa.com, 2018. [Online]. Available: https://www.escueladeescrituracreativa.com/gramatica/como-puntuar-dialogos-algunas-claves/#:~:text=La%20acotaciÃşn%2C%20tambiÃľn%20llamada%20inciso,una%20acotaciÃşn%20o%20no%20tenerla.

[18] A. P. G. Gutiérrez, "El diÁlogo," web.archive.org, 2009. [Online]. Available: https://web.archive.org/web/20180425184647id_/http://www.eumed.net/rev/cccss/04/apgg2.pdf

[19] "Diccionario panhispánico de dudas - raya," www.rae.es, 2005. [Online]. Available: https://www.rae.es/dpd/raya

[20] "Diccionario panhispánico de dudas - guión," www.rae.es, 2005. [Online]. Available: https://www.rae.es/dpd/guion

[21] "Tesoro de los diccionarios históricos de la lengua española - anáfora," www.rae.es, 2005. [Online]. Available: https://www.rae.es/tdhle/anÃąfora

[22] "Linguistic features," spacy.io, 2023. [Online]. Available: https://spacy.io/usage/linguistic-features

[23] J. Steinberger, M. Poesio, M. A. Kabadjov, and K. Ježek, "Two uses of anaphora resolution in summarization," www.sciencedirect.com,

2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457307000428

[24] G. Barlacchi and S. Tonelli, "Ernesta: A sentence simplification tool for children's stories in italian," link.springer.com, 2013. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-37256-8_39

[25] "Openai," openai.com, 2023. [Online]. Available: https://openai.com

[26] "Youchat," https://you.com, 2023. [Online]. Available: https://you.com

[27] "Spacy spanish," spacy.io, 2023. [Online]. Available: https://spacy.io/models/es

[28] J. J. Millas, "'mi pierna derecha' (cuento inédito)," elpais.com, 2008. [Online]. Available: https://elpais.com/diario/2008/10/14/cultura/1223935209_850215.html