CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Matemáticas e Informática

Universidad Politécnica de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

TRABAJO FIN DE GRADO

Desarrollo de aplicación web para proponer texto en lectura fácil

Autor: Clara Cerrato Garrido

Director: Mari Carmen Suárez de Figueroa Baonza

MADRID, JUNIO 2018

Resumen

Esta memoria contiene toda la información correspondiente al proceso de desarrollo de la aplicación del trabajo fin de grado: "Desarrollo de aplicación web para proponer texto en lectura fácil".

El trabajo fin de grado consiste en el desarrollo de una aplicación web que analiza un texto, escrito en **castellano** en formato libre,y tras el análisis del mismo recomienda una oración alternativa que cumple las directrices de la metodología de lectura fácil [1.1.2].

Entendemos como lectura fácil un metodología para adaptar y escribir un texto de la forma que sea sencillo de comprender por personas con dificultades lectoras, ya sean transitorias o permanentes. La elaboración de materiales de fácil lectura beneficia a personas con discapacidad intelectual [1.1.1] y competencias lingüísticas o lectoras limitadas.

Abstract

This memory contains the process developed for the final degree project called "Design of an application web to suggest a text Easy-to-read".

The final degree project consists on the develpoment of an application web which analyses a text writen in Spanish. This application web will suggest us an alternative sentence which fulfills the rules and the methodology of Easy-to-read [1.1.2].

Easy-to-read is a method that converts a text that is hard to understand for people with reading disabilities. In addition, this methodology also helps people with intellectual disabilities [1.1.1] and people with basic knowledge of a langueage.

Índice general

Capítulo 1

INTRODUCCIÓN

La finalidad de este capítulo es proporcionar una visión general de este proyecto, explicando su contexto, motivación y objetivos planteados.

1.1. Contexto

Este proyecto se enmarca dentro de la accesibilidad de la información, concretamente en la metodología de escritura de un texto para satisfacer las necesidades de personas con discapacidad intelectual y aquellas personas con dificultades a la hora de comprender un texto.

1.1.1. Discapacidad Intelectual

La discapacidad intelectual [22] es un término que se emplea para describir aquellas personas que tienen limitaciones tanto en el funcionamiento cerebral, como en las conductas adaptativas. Estas limitaciones se muestran a la hora de leer, hablar, caminar y actividades cotidianas, lo que puede ocasionar que un niño se desarrolle y aprenda de una manera más lenta o diferente. Las causas de la discapacidad intelectual puede ser por una lesión, enfermedad o un problema cerebral.

Una de las principales caracerísticas de la discapacidad intelectual es su heterogeidad. Aunque se trate del mismo síndrome, las personas tendrán distintos grados en las diferentes capacidades, las cuales son: memorias, lenguaje, abstracción, orientación, capacidad de razonamiento y capacidad de aprendizaje.

La discapacidad intelectual de un individuo no es una entidad fija e incambiable. Ya que se va modificando por el crecimiento y desarrollos biológicos del individuo y por la disponibilidad y calidad de los apoyos que recibe. En una interacción constante y permanente entre el individuo y su ambiente.

La heterogeneidad y su evolución a lo largo de los años de la persona causa la dificultad de fijar unas soluciones para facilitar la accesibilidad a un mayor número de personas, por ello, algunas de las soluciones se dan para un déficit en concreto.

En la actualidad hay diferentes soluciones para satisfacer las necesidades de las personas con discapacidad intelectual, entre ellas se encuentra la lectura fácil, aunque es una solución parcial enfocada para facilitar la comprensión de un texto.

1.1.2. Lectura fácil

La lectura fácil [19] surge como una forma de comprensión lectora. Su función es facilitar el acceso a la información, ya que todos queremos y necesitamos información. Sin embargo las personas con dificultades de aprendizaje se enfrentan al desafío de obtener la información que ellos necesitan.

Hablar de accesibilidad a los contenidos escritos significa no solo hablar de acceso a la literatura, los periódicos o las enciclopedias y libros de texto, sino también a la legislación, los documentos administrativos, los informes médicos, los contratos y cualquier otro texto de la vida cotidiana.

La lectura fácil es una manera de hacer la información más accesible para la gente con discapacidades intelectuales, pero también se trata de tener las siguientes caracteristícas:

- Hacer la información más sencilla
- Facilitar la comprensión de la información
- Emplear palabras e imágenes sencillas

No hay ninguna definición legal de como tiene que ser la lectura fácil. Sin embargo hay unos estándares mínimos. La Federación Internacional de Asociaciones de Bibliotecarios y Bibliotecas (IFLA) publicó en 1997 las Directrices para materiales de lectura fácil [18]. En este documento se recogen dos ideas distintas: la adaptación lingüística de un texto que lo hace más fácil de leer por su formato y adaptación que permite una lectura y comprensión más sencilla.

La lectura fácil no solo trata el contenido de la información, sino también las ilustraciones y la maquetación. Estos temas también son tratados en en las directrices para materiales de lectura fácil [18].

La IFLA establece tres niveles según la forma en la que se puede presentar la información:

1. Abundancia de ilustraciones y texto escaso de una complejidad sintáctica y lingüística baja
2. Vocabulario y expresiones de la vida cotidiana, acciones sencillas de seguir e ilustraciones
3. Un texto más largo, con palabras menos usuales y muy pocas ilustraciones

Debido a las características descritas que debe cumplir la información la metodología de lectura fácil, se pueden cubrir las necesidades de más personas. Además de las personas con discapacidad intelectual, en la Figura 1.1 se muestra a las diferentes personas que podrían ser beneficiadas cuando se emplea la metodología de lectura fácil.

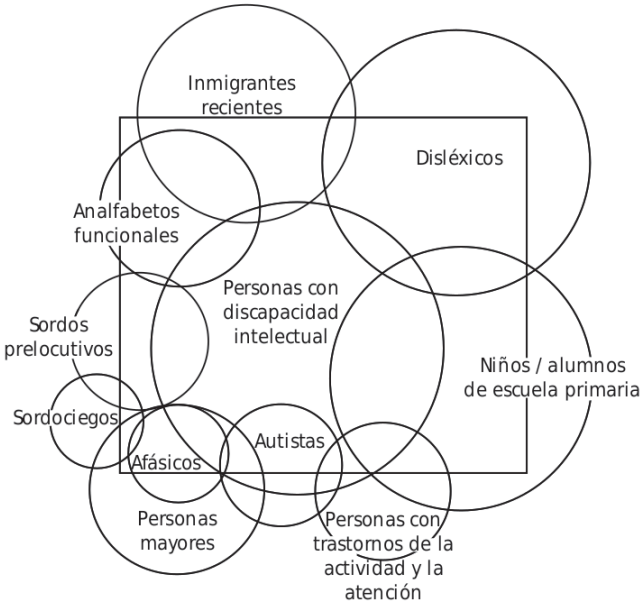

Figura 1.1: Usuarios de lectura fácil

Como se puede observar la gran mayoría son personas con discapacidades intelectuales[22], pero no son los únicos, también se pueden beneficiar casi cualquier persona con dificultades de comprensión e incluso niños.

1.2. Motivación y objetivos

Luego la **motivación** de este trabajo es lograr la accesibilidad de la información para todo el mundo, ya que es un derecho, y no siempre se cumple para las personas con dificultades de comprensión. Este proyecto tratará la mejora de la información de un texto, de manera que este escrito de forma clara, simple y cumpla las pautas de lectura fácil.

El **objetivo** de este proyecto es crear una aplicación web que permita al usuario obtener una alternativa a su texto, siempre que este no cumpla la metodología de lectura fácil.

La forma en que el usuario podrá analizar su texto será mediante un recuadro en una página web, donde se podrá escribir libremente texto en castellano. Al finalizar

de escribir, se presiona el botón de facilitar y se le mostrará la línea que no cumple alguna de las reglas de lectura fácil y a continuación como debería ser la frase que si cumple estos requisitos.

Para el desarrollo de este proyecto suponemos de antemano que tenemos una herramienta de otro trabajo fin de grado, la cual detecta si esa oración cumple las directrices de lectura fácil, dando una nota global del texto. Teniendo que reglas no se cumplen, nuestro trabajo consiste en facilitar un texto, es decir, realizar las transformaciones necesarias para que dicha oración satisfaga la metodología de lectura fácil.

Como requisitos del trabajo se han pautado:

- Estudio del lenguaje natural
- Estudio del arte
- Desarrollo aplicación web
- Estudio de la transformación de las oraciones
- Implementación del apartado anterior
- Unificación de la aplicación web y el programa
- Realización de pruebas del software durante todo el desarrollo, así como al finalizar cada módulo
- Redactar memoria final

Dentro de la realización de este trabajo, también se incluye el estudio y el análisis de la metodología de lectura fácil. Así como conocer cuales son los principales problemas que encuentran las personas con dificultades a la hora de entender un texto. Esto nos ayudará para lograr que nuestra propuesta se adapte todo lo posible a la metodología de lectura fácil.

Capítulo 2

ESTADO DEL ARTE

En este capítulo se realizará un estudio sobre las herramientas que se encuentran en la actualidad según nuestras necesidades para el desarrollo del proyecto

2.1. Lenguajes de programación

Un lenguaje de programación es un lenguaje formal definido como una secuencia de instrucciones ordenadas correctamente, las cuales permiten escribir un programa entendido por el computador. A la hora del desarrollo del trabajo fin de grado se han tenido en cuenta los siguientes lenguajes de programación:

2.1.1. Java

Java es un lenguaje de programación orientado a objetos,siendo uno de los lenguajes de programación más extendidos en el mundo y que actualmente se encuentra bajo la licencia de GPL de GNU.

Historia

Java comenzó como un proyecto llamado *Roble* por James Gosling en junio de 1991, surgió para implementar una máquina virtual y un lenguaje con notación tipo C, pero con mayor uniformidad y simplicidad. Gosling tenía como objetivo que una vez se escribiera el programa, fuera ejecutado, de esta manera los tiempos de ejecución libres en diferentes plataformas.

Java se puede emplear para le desarrollo de aplicaciones independientes, también permite el desarrollo de applets, las cuales son unas aplicaciones que se ejecutan dentro de un navegador cuando es cargada en una página HTML en un servidor web.

Gosling tenía claras cinco características que debía cumplir este lenguaje de programación:

Características

- Emplear programación orientada a objetos
- Permitir que se ejecute el mismo programa en múltiples sistemas operativos
- Tener un soporte incorporado para emplear redes de computadores
- Fácil manejo, seleccionado lo que él consideraba buenas características de otros lenguajes de programación orientados a objetos

2.1.2. Python

El lenguaje de programación Python[24] es un lenguaje interpretado, orientado a objetos y de alto nivel con semántica dinámica.

Historia

Python fue creado de la mano de Guido van Rossum en 1986, el cual surge como necesidad en el proyecto Amoeba. Se trataba de un sistema operativo distribuido, y necesitaban de un lenguaje orientado a objetos, que fuera sencillo de usar y capaz de tratar tareas dentro de la programación que realizaba en Unix empleando C. Guido comenzó desarrollando un lenguaje de *scripting* simple que poseía las mejores caracteísticas de ABC, pero siendo capaz de manejar excepciones e interactuar con Amoeba. A partir de este momento se complementa el lenguaje de diferentes maneras, entre ellas el lanzamiento de herramientas de programación funcional.

Desde la primera versión creada, Python 1.0, en 1994 hasta versión actual, abril 2018, Python 3.6.3, se han realizado incorporaciones de maneras muy diferentes como la toma de gran parte de las características del lenguaje funcional Haskell a la unificación de los tipos en Python, y las clases.

Características

Python se trata de un lenguaje multiparadigma, es decir, permite varios estilos a la hora de programar: programación orientada a objetos, programación imperativa y programación funcional.

Python emplea tipos dinámicos, lo cual lo hace muy atractivo para un rápido desarrollo de aplicaciones, así como emplearlo como *scripting*, es decir, soporta programas escritos en tiempo de ejecución, esta función es propia de los lenguajes interpretados. El uso de tipo dinámico y el conteo de referencia para la administración de memoria hace que es sea un lenguaje rápido por no necesitar una compilación.

En cuanto a sintaxis que emplea Python, es simple y fácil de aprender, ya que dicho lenguaje trata de enfatizar la legibilidad. Python admite módulos y paquetes, fomentando la modularidad del programa y la reutilización de código. Además,

existen muchas librerías que podemos importar a los programas para tratar temas específicos.

2.2. Tecnología y herramientas desarrollo web

2.2.1. HTML

HTML [5, 20] es un lenguaje de marcado de hipertexto, el cual se escribe con elementos, los cuales están construidos por etiquetas,contenido y atributos. HTML es un lenguaje que interpreta un navegador web para mostrar los sitios o aplicaciones web tal y como estamos acostumbrados.

El lenguaje HTML está limitado a la hora de aplicarle forma a un documento. Esto se debe a que fue concebido para otros usos, distintos a los actuales. Estos diseños han causado a menudo problemas en las páginas a la hora de su visualización en distintas plataformas. Para solucionar estos problemas los diseñadores han utilizado técnicas como la utilización de tablas transparentes, etiquetas que no son estándares de HTML y otras.

2.2.2. CSS

CSS [20] es un lenguaje que surge de la mano de *World wide web consortum (W3C)*, este organismo impulsó su estandarización a mitad de la década de los 90s.

Css se emplea para describir las propiedades de los estilos, generalmente, de un documento HTML, también es compatible con XML o SVG.

La idea más importante detrás de las hojas de estilo CSS es la separación del contenido de su presentación, es decir, de su aspecto visual. En las páginas web el lenguaje HTML se usa para estructurar el contenido semánticamente (títulos, subtítulos, texto, etc.) y CSS para la maquetación y estética del mismo. Este principio es fundamental por muchos motivos. Un ejemplo claro es el diseño adaptativo (responsive design): poder adaptar el mismo contenido a diferentes dispositivos. Es decir, que una misma página web se puede visualizar de una manera diferente en un PC que un móvil, optimizada para cada caso.

2.2.3. Php

PHP [9, 20, 28], acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de *scripting* de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender.

El objetivo principal de este lenguaje es permitir a los desarrolladores web escri-

bir dinámica y rápidamente páginas web generadas. También se puede emplear para enviar información a una base de datos.

2.2.4. Django

Definición *Framework* Web

El término *framework* hace referencia a una seria de motores para crear sitios web, el cual permite que el programador se centre en los diferentes módulos que componen una aplicación web, sin tener que emplear mucho tiempo en la comunicación de los diferentes módulos que se han desarrollado.

La misión principal de un *framework* Web es proveer la infraestructura de programación centrada en las aplicaciones, de manera que el programador se puede centrar en la escritura del código limpio y sencillo de mantener. Django pertenece a una nueva generación de framework Web.

Django

Django[13] es un *framework* para aplicaciones web de código abierto, escrito en Python. Es un conjunto de bibliotecas y herramientas que se permiten la creación de sitios web. Django permite al usuario construir sitios web dinámicos en corto periodo de tiempo.

Una de las metas de Django es facilitar la creación de sitios webs complejos. Django pone énfasis en la reutilización, la conectividad, el desarrollo rápido y el principio "*No te repitas*".

2.3. Bases de datos

En el desarrollo web es imprescindible el empleo de una base de datos, también se conocen como SGBD o sistema de gestión de bases de datos. Las bases de datos son un pilar fundamentar de cualquier aplicación web interactiva, ya que es donde se almacenarán de forma ordenada los datos que el usuario introduzca en la aplicación.

El objetivo fundamental de una base de datos es la administración de la información de una forma clara y ordenada.

Tanto Djando como Php son compatibles con numerosos tipos de datos. Aunque a la hora de la elección de una base de datos para el desarrollo de la página web solo se han tenido en cuenta MySql y PostSql, ya que son los dos tipos de bases de datos que se conocían. A coninuación se muestra un breve resumen de ambas con algunas de sus características.

2.3.1. MySql

Mysql [7, 20] es un sistema de gestión de bases de datos relacionales desarrollado por Oracle y Microsoft SQL Server, licenciado bajo la GPL de la GNU. La gestión de bases de datos se realiza con multihilo, multiusuario, de una forma rápida y segura.

La gestión y optimización de los accesos a disco en MySql viene dado por su arquitectura, ya que el motor de almacenamiento es independiente y transparente a los usuarios.Además Mysql está diseñado para permitir solicitudes simples desde una base de datos. A continuación veremos algunas características más concretas sobre este sistema de gestión.

- El lenguaje Sql tiene un amplio subconjunto
- Gran variedad de plataformas y sistemas
- Soporta diversos lenguajes de programación
- Portabilidad entre sistemas
- Seguridad de los datos

2.3.2. PostgreSQL

PostgreSql [10] es un sistema de bases de datos relacional de objetos que utiliza y amplía el lenguaje SQL. Postgre se inicia en un proyecto desarrollado por la universidad de California en 1986.

Postgre cuenta con una aquitectura comprobada,con fiabilidad, integridad de datos entre algunas de sus características. También se destaca por su sistema objeto-relacional que hereda las características de la orientación a objetos.

Algunas de las características principales son:

- Variedad en los tipos de datos
- Paralelización de las consultas
- Declaración de funciones propias
- Seguridad de los datos

Desarrollo Web

A continuación en la tabla 2.1 se muestran las tecnologías y lenguajes de programación que se han empleado finalmente para el desarrollo de la aplicación, así como su finalidad de uso. Tambíen se especifica el tipo de base de datos que se empleará para la misma.

Tecnologías	Descripción
Html	Determinará la estructura que sigue la información en la página web.
Css	Se empleará para el diseño y maquetación de la página HTML
Php	Enviará la información de la página web a la base de datos
MySQL	Enviará la información de la página web a la base de datos

Cuadro 2.1: Resumen herramientas web empleadas

2.4. Procesamiento del lenguaje natural

El lenguaje natural (NL) es el medio que utilizamos los seres humanos de forma cotidiana para comunicarnos [14]. Este se ha ido perfeccionando a partir de la experiencia, por ello en la actualidad se puede emplear para analizar situaciones complejas y razonar sutilmente.

Herramientas de Procesamiento de lenguaje natural(NLP)

En la actualidad muchos programas procesan con gran facilidad información estructurada, es decir, aquellos datos que tienen estructura y su significado es único y completo.

Sin embargo, más del 90 % de la información que se emplea en el mundo informático es información no está estructurada, cuya formato es texto y documentos en diferentes lenguas [6]. Para la comprensión de estos textos es necesario que se realice una interpretación correcta y realizar su relación táctica con la realidad. El desarrollo de dichos sistemas ha sido un reto para la lingüística e informática desde hace muchos años.

A la hora del desarrollo de herramientas capaces de analizar el lenguaje natural, se ha subdividido en diferentes tareas o niveles de procesamiento menos complejas, con la finalidad que sean abordadas por especialistas en los diferentes niveles:

- Nivel fonológico: es el nivel que trata el reconocimiento de sonidos, cuya finalidad es la conversión a palabras. Para un funcionamiento correcto es necesario el conocimiento y un algoritmo de fonemas
- Nivel morfológico: estudia la estructura interna de las palabras o definiciones y clasifica cada una de las unidades, las palabras a las que da lugar y la formación de nuevas palabras. Para ello se debe analizar los diferentes componentes de las palabras, sus monemas,su significado y la propia gramática de la palabra.

- Nivel sintáctico: estudia las formas en las que se combinan y relacionan las palabras para obtener estructuras comunicativas.
- Semántica: estudia los significados, sentido o interpretación de los símbolos. El desarrollo de analizadores semánticos presenta problemas debido a la ambigüedad de los términos dado que se extrae el contenido literal.
- Pragmática: estudia el modo en el que la interpretación del significado se ve influenciado por el contexto. Adapta el nivel semántico al significado real, ya sea al contexto o al uso adaptado.

Debido a la riqueza del lenguaje, el análisis de una sentencia tiene gran dificultad. La ambigüedad es una de las grandes limitaciones que se encuentran en cada uno de los niveles mencionados anteriormente. La consecuencia de esto es la necesidad de un gran número de reglas y estructuras para obtener cierta calidad en los sistemas. A pesar de ello encontramos sistemas eficaces, aunque con ciertos inconvenientes como el tiempo de procesamiento.

El conjunto de nuevas tareas aumenta según surgen nuevas necesidades, intereses o tecnología. Por consecuencia la lista de sistemas de procesamiento de lenguaje natural tampoco deja de crecer.

A continuación se muestra cuales son las herramientas de procesamiento de lenguaje natural, de castellano, que se han estudiado según los distintos niveles que deseábamos tratar en el desarrollo de nuestro proyecto:

2.4.1. FreeLing

Feeling[23] es una librería de código abierto para el procesamiento de lenguajes, la cual proporciona una amplia gama de funciones de análisis para varios idiomas, entre ellos el castellano.

El proyecto FreeLing fue creado en 2003 y actualmente está dirigido por Lluís Padró como un medio para poner en disposición de todo el mundo los resultados de la investigación de procesamiento de lenguaje natural en la Universidad Politécnica de Barcelona.

Se estructura como una librería y puede ser llamada desde cualquier aplicación de usuario que quiera realizar análisis del lenguaje. Está desarrollada en C++, altamente modular y con manuales tanto técnicos y de usuario.

La versión actual proporciona distintas funciones de análisis de lenguaje en castellano, las cuales se nombran a continuación:

- Segmentación de frases
- Tokenizador

- Reconocimiento de entidades
- Etiquetado morfosintáctico: EAGLES [16]
- Analizador sintáctico superficial.
- Identificador de la lengua del texto

Al introducir el texto que se desea analizar, se realiza una tokenización de cada palabra de la oración y se la asigna a cada una las siguientes etiquetas:

- **form**: token
- **lemma**: devuelve el origen de la palabra
- **tag**: etiqueta gramátical del token, empleando el sistema de etiquetas EAGLES [16]
- **ctag**: etiqueta gramátical
- **pos**: posición de la palabra (sustantivo, verbo, determinante)
- **type**: tipo de palabra según su posición
- **gen**: género de la palabra, si lo tiene
- **num**: número de la palabra

2.4.2. Stanford CoreNLP

Stanford CoreNLP [15] proporciona un conjunto de herramientas para el procesamiento del lenguaje natural en diferentes idiomas, entre ellos el castellano.

Se trata de un proyecto llevado a cabo por la Universidad de Standord en California (EEUU). En la actualidad el grupo Stanford NLP se encarga de su mantenimiento y actualización.

Se estructura como una librería en Java. Sin embargo, existen implementaciones para que se pueda utilizar en diferentes lenguajes de programación. Stanford CoreNLP está diseñado para ser flexible y extensible, ya que con una única opción pueden habilitarse y deshabilitarse herramientas. CoreNLP integra muchas herramientas, las cuales en castellano son:

- Tokenizador
- Separación de frases
- Etiqutador POS a cada token
- Reconoce entidades nombradas
- Análisis sintáctico constituyentes

2.4.3. NLTK

NLTK (natural Language Toolkit) [27] es una librería de Python orientada al aprendizaje e investigación del procesamiento natural u otras tareas relacionadas como la inteligencia artificial y el *machine learning*, entre otras.

NLTK incorpora 50 corpus con datos de entrenamiento de sus algoritmos, los cuales también se emplean para la realización de pruebas sobre los programas. Hay que destacar que no posee un soporte multilenguaje como base, luego puede ser entrenado a partir de diferentes corpus en idiomas distintos.

NLTK tiene como herramientas básicas:

- Tokenizador que emplea expresiones regulares para obtener los tokens del texto
- Etiquetas modosintácticas de Penn Treebank [8]
- Análisis sintáctico, el cual está entrenado con el corpus ACE[21]
- Reconoce entidades nombradas, entrando también con el corpus ACE

2.4.4. IXA Pipes

IXA Pipes[25] se trata de un conjunto modular de herramientas para el procesamiento del lenguaje natural. Está desarrollado en Java para el procesamiento del lenguaje natural basado en el lenguaje automático.

Ofrece anotaciones lingüísticas sólidas y eficientes para su empleo tanto en la investigación de profesionales expertos como para aquellos que no pertenecen a la IXA NLP de la Universidad del País Vasco.

Las tuberías de IXA se pueden emplear para exportar su modularidad, así como para seleccionar o cambiar algunas de sus componentes.

Las herramientas que nos proporciona son:

- Segmentación de oraciones
- Tokenizador
- Reconocimiento de entidades nombradas
- Etiquetado morfosintáctico
- Desambigüación léxica
- Analizador sintáctico
- Resolución de la correferencia

2.4.5. Gate

GATE (*General Architecture for Text Engineeting*)[1] es una arquitectura y aplicación desarrollada por la Universidad de Sheffield para la elaboración e integración de aplicaciones es tecnologías del lenguaje. GATE ejecuta en secuencias de módulos de procesamiento lingüístico para ejecutar todo tipo de tareas para el procesamiento del lenguaje. Gate Las herramientas básicas de Gate son:

- Segmentación de oraciones
- Tokenizador
- Reconocimiento de entidades nombradas
- Etiquetado morfosintáctico
- Analizador sintáctico
- Resolución de la correferencia

2.4.6. Spacy

Spacy [26] es una librería para el procesamientos de lenguaje natural, es de código abierto escrita en Python y Cython, publicada bajo licencia del instituto tecnológico de Massachusetts. Spacy a diferencia de las herramientas que hemos nombrado anteriormente, no se desarrolla para un ámbito académico o de investigación, si no para el uso de producciones reales. En la actualidad ofrece modelos estadísticos de redes neuronales para más de 20 idiomas, entre ellos el castellano. Algunas de las características son:

- Analizador sintáctico más rápido del mundo
- Reconocimiento de entidad nombrada
- Tokenización no destructiva
- Soporte para más de 20 idiomas
- Modelos estadísticos pre-entrenados y vectores de palabras
- Integración fácil de aprendizaje profundo
- Etiquetado de parte del discurso
- Análisis de dependencia etiquetado
- Segmentación de oraciones basadas en sintaxis
- Visualizadores integrados para sintaxis y NER
- Cómoda asignación de cadena a hash

- Exportar a matrices de datos numpy
- Eficiente serialización binaria
- Empaquetado y despliegue de modelo fácil
- Velocidad de vanguardia
- Exactitud robusta y rigurosamente evaluada

Resumen herramientas NLP

Tras un estudio de las anteriores herramientas mencionadas, todas ellas tienen muchas funciones que podrían resultar interesante, pero algunas no realizan esas funciones para el procesamiento del lenguaje escrito en **castellano**, por ello se ha realizado un resumen de las funciones que nos proporciona cada uno de ellos.

Tecnologías	Freeling	Core NLP	NLTK	IXA pipes	
Segementación de frase	X	X		X	
Tokenización	X	X	X	X	
Etiquetación morfosintáctica	X	X	X	X	
Lematización	X	X		X	
Desambigüación léxica	X			X	
Identificación de entidades	X	X	X	X	
Clasificación de entidades	X	X	X	X	
Sintaxis superficial	X		X	X	
Sintaxis constituyentes	X	X		X	
Sintaxis de dependencias	X	X		X	
Correferncia	X	X		X	

Cuadro 2.2: Resumen herramientas NLP

2.5. Otras herramientas y librerías

En este apartado se han incluido aquellas herramientas que no se podrían clasificar en ningunos de los apartados anteriores.

2.5.1. PyTorch

PyTorch [11] es un paquete de Python que proporcionada una programación de tensores, empleados para realizar cálculos numéricos. Para acelerar la realización de las operaciones existe la opción de su ejecución en GPU, lo cual hace que sea una plataforma que proporciona una gran velocidad. Pytorch consta de una interfaz para la realizar redes neuronales, luego es más sencilla que algunas librerías que existen para python.

Una de las principales características es su diseño pensado para ser intuitivo, lineal en pensamientos y fácil de usar. Debido a bibliotecas para la aceleración de GPUs se máximiza la velocidad y a la hora de entrenar o ejecutar redes neuronales grandes o pequeñas es bastante rápido.

Pytorch consta de la apa Pythorch's Tensor que permite la escritura de nuevos modulos de redes neurolares o interactuar con la API.

2.5.2. Pattern

Pattern esa organizado en módulos que Pattern [12] es una librería de python que se emplea para la minería de texto, contiene herramientas de procesamiento de lenguaje natural y algoritmos de aprendizaje automático y análisis de redes, con un manejo sencillo. También emplea diferentes herramientas simultaneas para utilizar las páginas web como Corpus.

Pattern tiene diferentes módulos, entre ellos uno de español, cuya funciones son:

- Singularización y pluralización de un sustantivo
- Conjugador de verbos
- Añadir género y número a un adjetivo
- Análisis de oraciones

2.5.3. Gensim

Gensim [4] es un biblioteca de Python disenãda para extraer automáticamente los temas semánticos de un texto sin estructurar.

Los algoritmos en gensim análizan y descubren la estructura semántica de los documentos, esto lo realizan estudiando patrones estadísticos de co-ocurrencia de las palabras dentros de un corpues de capacitación. Estos algoritmos no están supervidos luego solo enecitan un cuerpo de documentos de texto sin formato.

Una vez que se encuentran estos patrones estadísticos, cualquier documentos de

texto se puede representar en la nueva representación semántica y consultar su similitud con otros documentos

2.6. Conclusiones

Finalmente se ha concluido que para el desarrollo de este proyecto la implementación de las reglas se realzizaá en Python, ya que su sintaxis es sencilla y consta de multiples librerías de lenguaje natural, y su instalación es sencilla.

Como analizador sintáctico se empleará Freeling debido a que la información que devuelve en para el castellano es muy completa y sin ningún error. Aunque no sea el más rápido en este caso se prefiere un análisis más exhaustivo aunque tarde un poco más.

Aunque como analizador sintáctico principal se usará Freeling no se descarta el emplear otra de las herramientas estudiadas en el apartado 2.4 para el anánisis de una forma concreta.

Capítulo 3

DISEÑO

3.1. Reglas

Debido a que la función de nuestro sistema es mostrar una posible alternativa a una oración que no cumple la metodología de lectura fácil, la parte central de la aplicación será como la facilitación del texto.

En la metodología de lectura fácil se tratan una serie de pautas, algunas de ellas tienen que ver con la maquetación y contenido del texto,pero también con imágenes, tamaños y más características de un texto. Para la realización de este proyecto solo se aplicacarán aquellas reglas que tienen que ver con el contenido del texto [18, 19].

El objetivo de este proyecto es la implementación de una aplicación web que nos proponga un texto que cumpla las directrices de lectura fácil, si el texto introducido no cumple alguna de ellas.

La transformación de cada regla se realiza de forma diferente, pero en un orden lógico. Debido a que en algunos casos es necesario alterar las posiciones de las palabras en la oración, esto podría ocasionar que otras reglas no se satisfagan. Por ello se ha considerado que aquellas que se deben realizar primero son las oraciones que solo necesiten la eliminación o añadir una o varias palabras.

3.1.1. Selección de reglas

Reglas de números

Referente al uso de números se encuentran diferentes pautas según su contexto y uso. En la tabla 3.1 se muestra el número de reglas que se le ha otorgado para la realización del programa, por lo tanto no corresponde a número establecido de forma oficial.

N Regla	Tipo	Carterística	Descripción
N1	Números	Cifras	Los números en cifras preferentemente
N2	Números	Teléfono	Los números de teléfono deben estar en forma estándar NNN NN NN NN
N3	Números	Fechas	Las fechas deberán estar de forma DD mes de AAAA, siendo DD y AAAA cifras
N4	Números	Romanos	Evitar el uso de números romanos
N5	Números	Dinero	Se debe hacer la separación de Euros y céntimos

Cuadro 3.1: Reglas sobre números y cifras

Reglas de léxico

En cuanto a las reglas que hacen referencia al léxico de la oración que se emplea en un texto, se han seleccionado aquellas que se mencionan en la tabla 3.2, la numeración de las reglas han sido definidas por nosotros.

N Regla	Tipo	Característica	Descripción
V1	Léxico	Siglas	No emplear el uso de siglas o acrónimos
V2	Léxico	Adverbios	Evitar adverbios acabados en -mente
V3	Léxico	Porcentajes	Evitar porcentajes

Cuadro 3.2: Reglas de léxico

Reglas Gramaticales

En cuanto las reglas de gramática se tratarán primero aquellas que solo necesiten de la eliminación o añadir uno o varios elementos, posteriormente se tratarán

aquellas que necesitan la alteración del orden de algunas palabras. El orden y las reglas seleccionadas son aquellas que aparecen en la tabla 3.3.

N Regla	Tipo	Característica	Descripción
G1	Números	Oraciones	Existencia de sujeto, preferiblemente por repetición o pronombre
G2	Gramática	Tipo oración	No hacer uso de las formas compuestas verbales
G3	Gramática	Estructura oración	Evitar el uso de la voz pasiva
G4	Gramática	Verbos	Evitar verbos modo subjuntivo
G5	Gramática	Tipo oración	Evitar las subordinadas adverbiales consecutivas y concesivas
G6	Gramática	Oraciones	No emplear las dobles negaciones
G7	Gramática	Estructura oración	La longitud de la oración se menor de 15 palabras

Cuadro 3.3: Reglas gramaticales

La principal idea en el diseño de las reglas ha sido permitir un desarrollo independiente entre ellas, con el fin que este trabajo pueda ampliarse en un futuro, añadiendo más transformaciones para las reglas que no se hayan logrado implementar.

3.1.2. Tratamiento y transformación de las reglas

Para realizar las transformación de las reglas que se han nombrado anteriormente en el apartado 3.1.1, será necesario el uso de Freeling para la identificación de los números.

Reglas de números

Para realizar la facilitación de una oración en la que aparecen números su planteamiento será igual para todas ellas. Se plantean dos posibilidades de resolución:

realizar un estudio de las librerías observando como tratan cada uno de los caracteres y en caso de no encontrar ninguna librería que realizará esta labor, se procederá a la implementación de un programa en el que se convierta cada literal en un número e ir sumando según aparecen más números.

Observamos que Freeling distingue según el contexto en distintos tipos de números y los etiqueta de la siguiente manera:

- pos= date → fechas,horas y siglos
- pos=number y ctag= Zm → dinero
- por=number y ctag= Z → resto de cifras

A continuación se detalla como se tratará cada una de las reglas de fácil lectura que hacen referencia a los números, así como la solución.

- **N1**: Los números deben aparecer en cifras

  Resolución: Se realiza el análisis con Freeling y siempre que se se trate de algunas de las etiquetas mencionadas en 3.1.2 se tomará la etiqueta lemma.

  Tratamiento: Una vez que se tiene el valor de la etiqueta lemma se tomará aquellas partes que pertenezcan a los números obteniéndolos ya en cifras.

- **N2**: Los números de teléfono con formato NNN NN NN NN.

  Resolución: Se empleará Freeling,y cuando se identifique la etiqueta pos=number se llamará a un función *fecha* que hemos implementado .

  Tratamiento: Se realiza la llamada a nuestra función *teléfono* que separará los números según la regla.

- **N3**: Las fechas deberán estar en formato DD mes de AAAA, donde DD y AAAA se escriben con cifras.

  Resolución: Se empleará Freeling,y cuando se identifique la etiqueta pos=fecha se guarda el lemma de dicha palabra, ya que Freeling lo convierte en formato [w:DD/MM/AAAA:HH.MM:am/pm] en la etiqueta lemma.

  Freeling solo identifica con la etiqueta con por=fecha si la fecha esta escrita en lenguaje natural, pero si se escribe en los siguiente formatos , los identifica como pos=number.

- DD/MM/AAAA o DD-MM-AAAA
- DD/MM/AA o DD-MM-AA
- DD/mes/AAAA o DD-mes-AAAA
- DD/mes/AAAA o DD-mes-AAAA
- DD/ENE/AAAA o DD-ENE-AAAA
- DD/ENE/AA o DD-ENE-AA

Tratamiento: Si es pos=fecha, se obtiene el resultado de la etiqueta lemma y con la función *fecha* se convierte al formato que nos indica la regla, donde el mes aparece escrito con palabras.

En cambio si se en la oración aparece en formato nombra en el planteamineto 3.1.2 llamaremos al programa *expandDate* que se ha implementado tratando los meses con un diccionario y devuelve la fecha en el formato dd de mes del aaaa.

- **N4**: Los números romanos deben escribirse en cifras.

  Resolución: Se empleará Freeling,y cuando se identifique la etiqueta pos=fecha en el lemma aparece el formato [s:sigo en números romanos]. Como no es lo que deseamos se ha implementado la función *romanos*.

  Tratamiento: Se llama a la función *romanos* que convierte cualquier número romano en cifra.

- **N5**: Se debe realizar separación entre Euros y céntimos.

  Resolución: Se empleará Freeling para la identificación de la aparición de la cifra y se llamara a la función *euros*.

  Tratamiento: Se llama a la función *euros* que separa la cifra hasta el punto donde se añade las palabras *Euros y con*, se incorporá la parte decimal, y a continuación la palabra céntimos.

Reglas léxicas

A continuación se detalla como se tratará y la solución dada para cada una de las reglas de fácil lectura que hacen referencia al léxico que se emplea en las oraciones.

- **V1**: No emplear el uso de siglas o acrónimos.

  Planteamiento: Identificar la sigla o acrónimo que aparece en la oración y posteriormente remplazarla por su forma extendida. Para la forma extendida se plantearon dos opciones, una empleando el word2vec entrenándolo con

texto que contengan las abreviaturas y su objetivo es que nos devuelva la forma completa escrita en lenguaje natural.

La otra posibilidad es la creación de un diccionario en Python con los acrónimos más comunes. Esta última será la cual implementaremos ya que tras probar word2vec, se observa que no devuelve el valor deseado y se descarta esta opción.

Resolución: Para la identificación de la sigla o el acrónimo no se ha podido emplear ningunas de las herramientas de procesamiento de lenguaje natural, ya que no devuelve ninguna etiqueta distintiva al tratarse de una sigla o acrónimo, luego el tratamiento y la identificación se realizarán con la función *siglas*.

Tratamiento: Para escribir las siglas o acrónimos en forma extendida se ha implementado un diccionario donde se encuentran escritas completamente. La función *siglas* buscará en la oración la sigla o acrónimo en el diccionario y la remplazará por su versión completa.

- **V2**: Evitar adverbios acabados en -mente.

  Planteamiento: Identificar el adverbio y cambiarlo por *de manera* más el mismo adverbio pero sin la terminación en mente.

  Resolución: Se empleará Freeling para identificación del adverbio, ya que devuelve en una de las etiquetas pos=.ªdverbƶ se realiza la llamada a la función *adverbio*.

  Tratamiento: La función *adverbio* extre el adverbio sin la terminación mente, lo elimina y lo sustituye por *de manera + adverbio*.

- **V3**: Escribir los porcentajes en lenguaje natural.

  Planteamiento: Freeling distingue cuando hay un porcentaje con las etiquesas ctag=ZP y pos= number, luego de esta manera identificaremos los porcentajes.

  Resolución: Se empleará Freeling,y cuando se identifique la etiquetas ctag=ZP y pos= number, llamaremos a la función *porcentajes* con la palabra a la que corresponden estos identificadores.

  Tratamiento: La función *porcentajes* llama a un diccionario que se ha creado donde se tranforman según la cantidad a su equivalente de manera escrita. Por ejemplo: 50 % → la mitad

Reglas gramaticales

Para las reglas referentes a la gramática del texto primero se plantea una solución. Después del estudio de ese planteamiento se propone la resolución que se empleará y finalmente como se realizará la facilitación de la oración.

- **G1**: Existencia de sujeto, preferiblemente por repetición o pronombre.

  Planteamiento: En el tratamiento de añadir un sujeto, se ha planteado el uso de diversas herramientas según nuestras necesidades.

  Se plantea el uso Textacy ya que consigue extraer los personajes principales del texto, así como las veces que aparece. Es cierto que devuelve un sujeto en concreto pero con un analizador semántico, se logrará nuestro objetivo.

  Por otro lado Freeling, nos analiza la forma y número del verbo, y de esta manera podríamos emplear un pronombre aunque no es lo más recomendable para lectura fácil .

  Resolución: Finalmente se empleará el añadir un pronombre ya que de esta forma nos aseguramos que sea más fiable.

  Tratamiento: Se analizará con Freeling el verbo de la oración, tanto su persona como número y según los resultados obtenidos se añadirá el pronombre que corresponde.

- **G2**: No hacer uso de las formas verbales compuestas.

  Planteamiento: Para la facilitación de las oraciones con formas verbales compuestas se propone mantener el tiempo, modo, persona y número del verbo pero en su forma simple con la librería pattern 2.5.2.

  Resolución: Se empleará Freeling para la identificación de los verbos, tanto de persona número, tiempo y modo. Aunque al tratarse de formas compuestas analiza el verbo auxiliar y posteriormente el verbo principal que se encontrará en el modo participio, esto nos lleva a que las características dadas no corresponden con como nosotros analizamos las formas verbales y se tengan que realizar cambios.

  Tratamiento: La función *compuestos* obtiene las caracterśticas del verbos y estas se cambian a la sintaxis que es capaz de interpretar Pattern. Una vez que obtenemos el verbo principal conjugado se sustituirá el verbo en forma compuesta por el devuelto por pattern.

- **G3**: Evitar el uso de la voz pasiva.

Planteamiento: En la facilitación de esta regla es necesario el cambio del orden de los complementos, para ellos se pone si existe complemento agente este pasará a ser el sujeto de la oración y el sujeto complemento del verbo.

Si no existe complemento agente el sujeto pasará a complemento del verbo y se añadirá un sujeto empleando la misma técnica que en el apartado 3.1.2 .

En ambos casos el verbor ser será omitido y se conjugará el verbo principal en el mismo tiempo, modo, persona y número que se encontraba el verbo auxiliar.

Resolución: Realizando el análisis sintáctico con Freeling detectaremos si existe el complemento agente y sujeto, se identificará el modo, tiempo, persona y número del verbo ser. Finalmente se realiza la llamada la función *pasiva* que se encargará de establecer el orden y conjugación del verbo.

Tratamiento: Se cambian las características recibidas del verbo para que Pattern realiza correctamente la tranformación y se reestructura la frase tal como se ha detallado en 3.1.2

- **G4**: Evitar verbos modo subjuntivo.

  Planteamiento: Para el estudio de este problema se ha empleado como apoyo el documento *EL subjuntivo: nuevas reglas para nuevas estrategias* [17], como ya se intuía sobre esta regla la facilitación del texto es necesario la realización de una nueva oración con el mismo significado aunque una estructura complentamente diferente.

  Esta regla se resolverá en el futuro en la continuación de este trabajo.

- **G5**: Evitar las subordinadas adverbiales consecutivas y concesivas.

  Planteamiento: se realiza un estudio sobre las oraciones subordinadas adverbiales concesivas [2] y tras un análisis se llega a la conclusión de que para el cumplimiento de esta regla su facilitación sería con una oración totalmente nueva.

  En cuanto a las oraciones subordinadas adverbiales consecutivas [3] se plantea que una manera de la facilitación sería, reordenar la frase, de manera que la parte consecutiva se encuentre al principio y las sentencias anteriores al nexo al final de la oración, unidas por la conjunción, ya que a la hora de realizar una análisis más en profundidad observamos que esto cubriría casos muy concretos

- **G6**: No emplear dobles negaciones.

Planteamiento: Se ha realizado un estudio de todos los casos en los que se emplea la doble negación en castellano, y no han podido idenficar patrones para su implementación y tranformación. Esto se debe a que para la transformación de una doble negación se debe construir una nueva oración, por lo tanto actualmente esta regla no se ha implementado.

- **G7**: La longitud de la oración ser menor de 15 palabras.

  Planteamiento: En el estudio de las herramientas de procesamiento de lenguaje natural 2.4 se encontró que con NLTK o con el desarrollo de un modelo en el que se entrenará introduciendo un texto y a continuación su resumen.

  Debido al tiempo que se requería para la preparación de los datos para el entrenamiento del modelos este apartado se desarrollará en un futuro.

3.2. Diseño aplicación

Para realizar el desarrollo de la aplicación web se ha empleado Html, Css para el diseño. También se ha hecho uso de Php para la conexión del programa desarrollado en Python con la aplicación web. Como base de datos se ha empleado MySql, donde se almacenará tanto los datos que recibe la aplicación como los devueltos por el programa.

En la aplicación existen cuatro ventanas,a las cuales se puede acceder navegando hacia abajo o pinchando directamente en cada una de las secciones que aparecen en el menú situado a la derecha. A continuación se detalla cada una de las ventanas y sus funciones.

3.2.1. Pestaña inicio

Esta página es la primera vista que tiene el usuario al acceder a nuestra aplicación. En ella el usuario tiene un menú en la parte superior izquierda donde puede dirigirse a tres pantallas diferentes según lo que desea hacer, el inicio, que es en la que se encuentra. El segundo botón que nos dirige a la sección 3.2.2,el según botón que nos redirige a la sección 3.2.3 donde explica las reglas implementadas y su número el cuarto botón donde se introducirá el texto a facilitar, que nos redirige a la sección 3.2.4

Figura 3.1: página inicio web

3.2.2. Pestaña ¿Qué es lectura fácil?

En esta ventana sirve para tener una idea de las características que tiene la lectura fácil. Se da una definición de lectura fácil así como sus beneficios. En la definición se explica también en que propiedades se centra esta metodología aunque nosotros solo trataremos el contenido del texto.

Figura 3.2: Lectura fácil

3.2.3. Pestaña Nuestra aplicación

En esta vista se explicará las reglas que nuestra aplicación es capaz de facilitar. El número que hay a la izquierda indica la cifra que habría que introducir a continuación de cada una de las reglas. En esta ventana se ha procurado que la explicación sea suficientemente clara.

Figura 3.3: Nuestra aplicación

3.2.4. Pestaña Facilitar un texto

En esta pestaña encontramos un cuadro de texto en el que se introducirá frase por frase y entre corchetes el número de reglas que no cumplen. Una vez que se hayan introducido todas las frases que deseamos le daremos a botón facilitar.

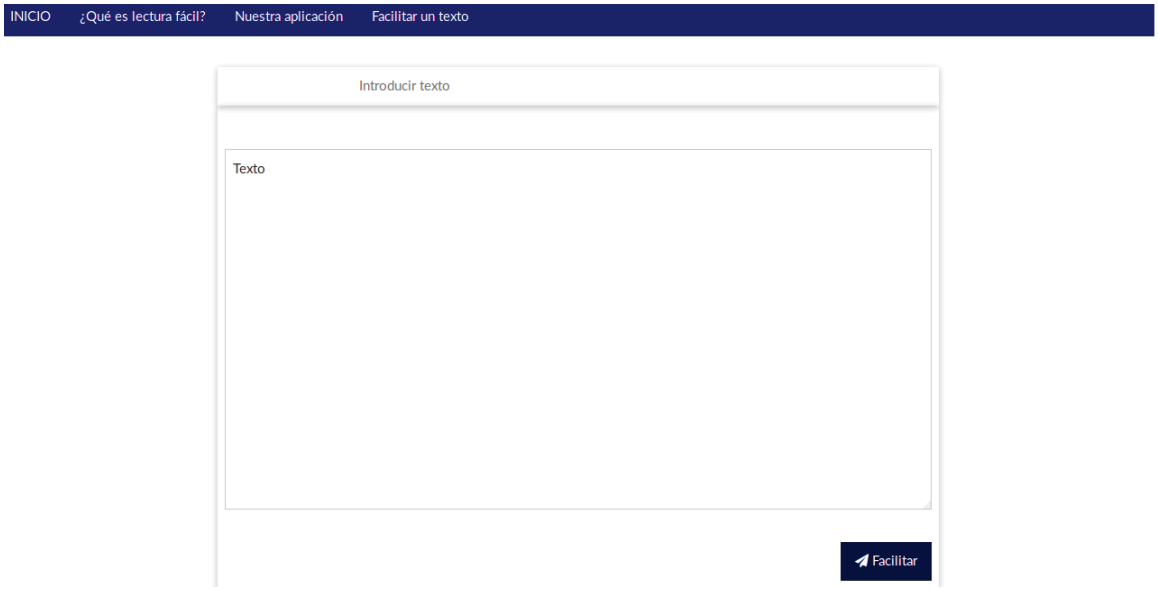

Figura 3.4: Facilitar un texto

3.2.5. Pestaña texto facilitado

A esta ventana solo llegaremos una vez que se haya dado al botón facilitar. En ella se nos mostrará frase por frase que se solicitó facilitar y a conitnuacón, la pro-

puesta de como será cumpliendo las directrices de lectura fácil.

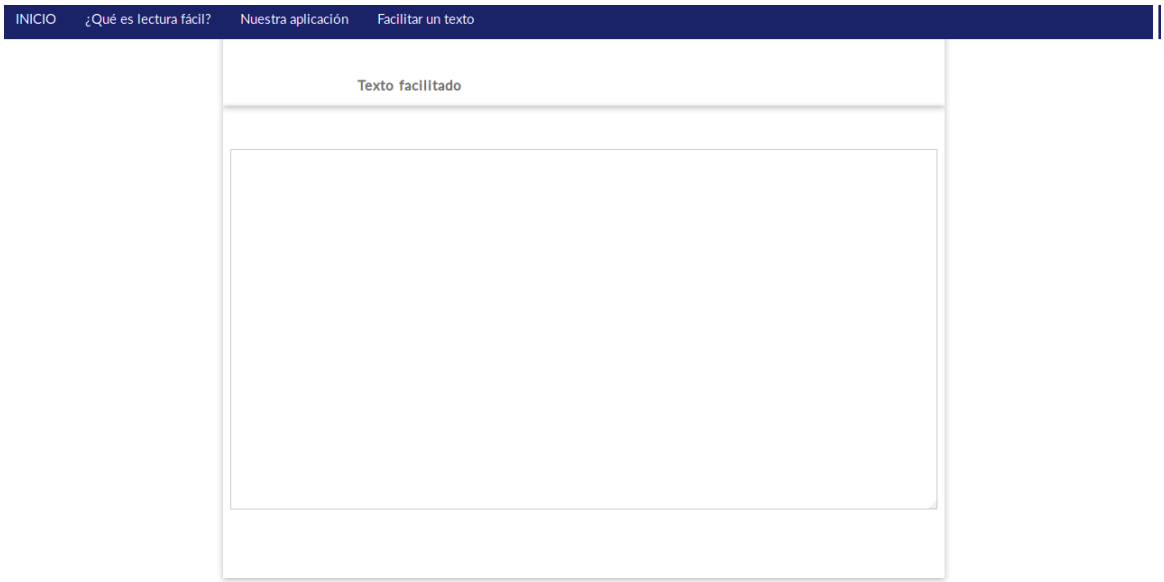

Figura 3.5: Texto facilitado

Capítulo 4

PRUEBAS REALIZADAS

Para la realización de las pruebas se va a emplear tablas con el mismos formato que la tabla 4.1 donde N regla será la el número que se le ha otorgado en el apartado 3.1.1. En la casilla Prueba se especificará exactamente la palabra o palabras que sobre las que se realizan las pruebas. En cuanto a las casillas de Oración introducida se trata de la consulta que realizamos al programa, así como la oración devuelta aquella que el programa que nos retorna. Finalmente en la última fila se marcará la satisfacción del resultado.

Las pruebas que se han realizado han sido según los criterios de cada regla, la casilla de aceptar, mejorar y rechazar se han valorado para el cumplimiento de la regla. Una correcta validación de las oraciones sería con un experto en lectura fácil o una persona con discapacidad intelectual.

N Regla	Prueba	
Oración introducida		
Oración devuelta		
Aceptar	✓Mejorar	Rechazada

Cuadro 4.1: Tabla pruebas

4.1. Pruebas reglas númericas

4.1.1. Regla N1 - Números en cifras

N1	millares, centenas, decenas	
El autobús tenía mil plazas, doscientos cuarenta y tres hombres y doce niños		
el autobús tenía 1000 plazas , 243 hombres y 12 niños		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.2: Pruebas números

N1	cifras con decimales: quience con diez	
El niño compró quince con diez kilos de manzanas		
El niño compró 15 con 10 kilos de manzanas		
Aceptar	✓Mejorar	Rechazada

Cuadro 4.3: Pruebas decimales

4.1.2. Regla N2 - Números de teléfonos

N2	123456789	
Mi número de teléfono es el mismo: 123456789		
Mi número de teléfono es el mismo: 123 45 67 89		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.4: Prueba teléfono

4.1.3. Regla N3 - Fechas

N3	día de mes del año	
El día de su billete es el treinta y uno de enero del dos mil dieciocho		
el día de su billete es el 31 de Enero del 2018		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.5: Prueba día de mes del AAAA

N3	día de mes del AAAA	
La fecha de envío del paquete es el diecinueve de mayo del 2017		
La fecha de envío de el paquete es el 19 de Mayo del 2017		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.6: Prueba día de mes del AAAA

N3	día de mes del AA	
La fecha de envío de el paquete es el diecinueve de mayo del 17		
La fecha de envío de el paquete es el 19 de Mayo del 2017		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.7: Prueba día de mes del AAAA

N3	d de mes del AA	
El día de la última entrega es el 1 de febrero del 18		
El día de la última entrega es el 1 de Febrero del 2018		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.8: Prueba dd de mes del AAAA

N3	dd-mm y dd/mm	
La fiesta será el 07-07 es decir el 07/07		
La fiesta será el 07 de Julio es decir 07 de Julio		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.9: Prueba dd de mes del AAAA

N3	dd-mm-AAAA y dd/mm/AAAA	
Lorca murió el 18-08-1936, lo que es lo mismo el 18/08/1936		
Lorca murió el 18 de Agosto del 1936 , lo que es lo mismo el 18 de Agosto del 1936		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.10: Prueba dd-mm-AAAA y dd/mm/AAAA

N3	dd-MES-AA	
El cambio del sistema fue el 1-DIC-17		
El cambio de el sistema fue el 1 de Diciembre del 2017		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.11: Prueba dd-MES-AA

N3	dd/MES/AA	
No estuvo disponible hasta el 4-OCT-18		
no estuvo disponible hasta el 4 de Octubre del 2018		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.12: Prueba dd-MES-AA

4.1.4. Regla N4 - Números romanos

N3	Unidaes	
En el siglo II se construyó el panteón de Roma		
En el siglo 2 se construyó el panteón de Roma		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.13: Prueba unidades siglo

N3	decenas sin referirse a siglo	
En la página XVIII aparece la introducción		
En la pagina 18 aparece la introducción		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.14: centenas sin referirse a siglo

N4	centenas y millares	
CCLXXIII, DLXXXIX, MDCCCLXXIII		
273 , 589 , 1873		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.15: Prueba centenas y millares

4.1.5. Regla N5 - Números refiriéndose a dinero

N5	Sin decimales	
Las manzanas cuestan 2 euros		
Las manzanas cuestan 2 euros		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.16: Prueba sin decimales

N5	Con decimales	
El billete cuesta 53,25 euros		
El billete cuesta 53 euros con 25 céntimos		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.17: Prueba con decimales

4.1.6. Más pruebas con números

En este apartado las pruebas que se realizan son para comprobar el funcionamiento de nuestro programa cuando Freeling detecta un contexto especifico, ya que a la hora de programarlas se ha podido olvidar algunos casos.

N5	Dinero en lenguaje natural	
El billete son cincuenta y tres euros		
El billete son cincuenta_y_tres_euros		
Aceptar	Mejorar	✓Rechazada

Cuadro 4.18: Prueba en contexto dinero en lenguaje natural

N5	Números referentes a kilos	
Compré tres kilos y medio de patatas		
compré 3 kilos y medio de patatas		
Aceptar	Mejorar	✓Rechazada

Cuadro 4.19: Números referentes a kilos

4.2. Pruebas reglas vocabulario

4.2.1. Regla V1 - Siglas y acrónimos

N1	AVE	
El AVE se inauguró hace 25 años		
El tren de Alta Velocidad Espanñola hace 25 años		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.20: Pruebas números

N1	PYME	
Las PYMES necesitan ayuda para su desarrollo		
Las pequeñas y medianas empresas necesitan ayuda para su desarrollo		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.21: Pruebas números

4.2.2. Regla V2 - Adverbios -mente

N1	adverbio	
El profesor lo dijo fuertemente		
El profesor lo dijo de manera fuerte		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.22: Pruebas adverbio

4.2.3. Regla V3 - Porcentajes

N1	cifras: 50 %	
El 50 % de la población		
El la mitad e la población		
Aceptar	✓Mejorar	Rechazada

Cuadro 4.23: Pruebas porcentaje 50 %

N1	cifras: 20 % y 80 %	
El 20 % sobrevive al virus, es decir muere un 80 %		
El 1 de cada 5 sobrevive a el virus , es decir muere un 1 de cada 5’		
Aceptar	✓Mejorar	Rechazada

Cuadro 4.24: Pruebas porcentaje 20 y 80

4.3. Pruebas reglas gramaticales

4.3.1. Regla G1 - Sujeto

N1	sujeto tercera persona singular	
fue a la biblioteca		
él fue a la biblioteca		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.25: Prueba tercera persona singular

N1	primera persona singular	
Ayer estuve en el parque		
ayer yo estuve en parque		
Aceptar	✓Mejorar	Rechazada

Cuadro 4.26: Prueba primera persona del singular

N1	segunda persona del plural	
Durante toda la noche bailasteis mucho		
durante toda la noche vosotros bailasteis mucho		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.27: Prueba segunda persona del plural

N1	tercera persona del plural	
estuvieron cenando en el restaurante nuevo		
ellos estuvieron cenando en el restaurante nuevo		
Aceptar	Mejorar	✓Rechazada

Cuadro 4.28: Prueba tercera persona del plural

4.3.2. Regla G2 - Verbo compuesto

N1	pretérito perfecto	
Carlos ha estado muy bien		
carlos estaba muy bien		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.29: prueba pretérito perfecto

N1	pluscuamperfecto	
Juan y tu habíais estudiado mucho en la biblioteca		
Juan y tu estudiabais mucjo en la biblioteca		
Aceptar	✓Mejorar	Rechazada

Cuadro 4.30: pluscuamperfecto

N1	futuro	
Mañana habremos caminado 4 kilómetros		
Mañana caminaremos 4 kilómetros		
Aceptar	✓Mejorar	Rechazada

Cuadro 4.31: futuro

N1	condicional	
Si habríamos corrido hubiéramos ido en coche		
Si corríeramos hubiéremos iríamos en coche		
Aceptar	Mejorar	✓Rechazada

Cuadro 4.32: futuro

4.3.3. Regla G3 - Voz pasiva

N1	sin complemento agente	
La mujer fue trasladada al hospital		
trasladaban a la mujer a el hospital		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.33: prueba sin complemento agente

N1	con complemento agente	
La habitación fue recogida por el hijo		
el hijo recoga la habitaciín		
✓Aceptar	Mejorar	Rechazada

Cuadro 4.34: prueba con complemento agente

N1	complemento agente posesivo	
El gato es alimentado por mi		
mi alimentan el gato		
Aceptar	Mejorar	✓Rechazada

Cuadro 4.35: complemento agente posesivo

Capítulo 5

LÍNEAS FUTURAS

Como se ha visto en el apartado 3.1.2 algunas de las reglas no se han podido implementar, ya haya sido por la falta de tecnología o el tiempo para realizar el entrenamiento de un modelo.

De aquí en adelante, la línea a seguir será en un proyecto de trabajo fin de máster, en el cuál se estudiarán métodos cognitivos basados en la inteligencia arficial así como las redes neuronales y redes neuronales profundas.

Se pretende realizar un estudio más detallado para conseguir sustituir a los sistemas basados en reglas, que requiren del estudio de datos para la identificación de las mismas, por un sistema que sea capaz de identificarlas y transformarlas, para así cubrir más reglas y todos los distintos casos que existen dentro de cada regla.

Sin embargo, una aproximación a través del entrenamiento de una red neuronal puede llegar a un mayor número de casos distintos. Esta aproximación al problema se puede ver ya en el mundo real en el caso de bancos y aseguradoras que ya están intentando o ya han conseguido sustituir los sistemas existentes basados en reglas para la detección de fraude por modelos basados en redes neuronales o en redes bayesianas.

Capítulo 6

CONCLUSIONES

Durante la realización de este trabajo se han adquirido los conocimientos básicos sobre el funcionamiento de una aplicación web, así como algunas de las tecnologías para su desarrollo.

Fundamentalmente este trabajo ha ayudado a adquirir un conocimiento más profundo sobre el lenguaje natural, ya que al realizar el estudio de las diferentes aplicaciones existentes muchas de ellas son de código abierto y se ha tratado de conocer mejor su funcionamiento e implementación. De esta forma se han sacado conclusiones más claras para continuar en un trabajo futuro así como un mayor aprendizaje del uso de la inteligencia artificial.

Tras el estudio de las aplicaciones, aunque en este proyecto no se hayan empleado, muchas de ellas se emplearán en un futuro en la continuación de este proyecto y también han sido de gran utilidad para ver el funcionamiento e implementación de procesador de lenguaje natural.

Cabe destacar que la realización de esta aplicación ha supuesto la adquisición de una visión mucho más amplia con el procesamiento del lenguaje natural, así como el análisis del déficit que existe en el análisis de los texto en castellano.

Finalmente, durante el desarrollo de este trabajo se han adquirido conocimiento que no se han visto durante la carrera y que serán de gran utilidad para el desarrollo en un futuro.

Bibliografía

[1] URL https://gate.ac.uk/sale/tao/split.html.

[2] *.Ejemplos de Oracin Subordinada Concesiva "*, . URL https://www.gramaticas.net/2012/07/ejemplos-de-subordinada-adverbial.html.

[3] *.Ejemplos de Subordinada Adverbial Consecutiva"*, . URL https://www.gramaticas.net/2012/07/ejemplos-de-subordinada-adverbial.html.

[4] *Gensim*. URL https://radimrehurek.com/gensim/intro.html.

[5] *Html*. URL http://www.w3schools.com/html/default.asp.

[6] *Informe sobre el estado de las tecnologías del lenguaje en España dentro de la Agenda Digital para España*. URL http://www.agendadigital.gob.es/tecnologias-lenguaje/BibliotecaimpulsotecnologiaslenguajeMaterial%20complementario/Informe-Tecnologias-Lenguaje-Espana.pdf.

[7] *MySQL Basics*. URL http://www.w3schools.com/sql/default.asp.

[8] *Penn Treebank*. URL https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html.

[9] *Php*. URL http://php.net/manual/es/preface.php.

[10] *postgresql*. URL https://www.postgresql.org/about/.

[11] *Pythorch*. URL https://pytorch.org/about/.

[12] *Pattern for Python*, 2012.

[13] Jacob Kaplan-Moss Adrian Holovay. *The definitive guide to Django*.

[14] Jaime Carbonell. *El procesamiento del lenguaje natural, tecnología en transición*. URL https://cvc.cervantes.es/obref/congresos/sevilla/tecnologias/ponenc_carbonell.htm.

[15] John Bauer Jenny Finkel Steven J. Bethard Christopher D.Manning, Mihai Surdeanu y David McClosky. *The Stanford CoreNLP Natural Language Processing Toolkit*, 2014. URL http://nlp.stanford.edu/pubs/StanfordCoreNlp2014.pdf.

[16] Linguistic Data Consortium. *ACE 2004 Multilingual Training Corpus*. URL https://catalog.ldc.upenn.edu/LDC2005T09.

[17] Javier García de María. *El subjuntivo: nuevas reglas para nuevas estrategias*.

[18] Revisin de Misako Nomura, Gyda Skat Nielsen, y Bror Tronbacke. Directrices para materiales de lectura fácil. *IFLA*, 2010. URL https://www.ifla.org/files/assets/hq/publications/professional-report/120-es.pdf.

[19] Óscar García Munoz. *Lectura fácil: métodos de redacción y evaluación*. Real patronato sobre discapacidad, 2012. URL http://www.plenainclusion.org/sites/default/files/lectura-facil-metodos.pdf.

[20] Robin Nixon. *Learning PHP, MySQL, Javascript, CSS, HTML5. A step by step guide to creating dynamic web sites.*, .

[21] Robin Nixon. *Learning PHP, MySQL, Javascript, CSS, HTML5. A step by step guide to creating dynamic web sites.*, .

[22] National Center on Birth Defects y Developmental Disabilities. *Facts About Intellectual Disability*. URL https://www.cdc.gov/ncbddd/actearly/pdf/parents_pdfs/IntellectualDisability.pdf.

[23] Lluís Padró. Freeling user manual. 2016.

[24] Luís Padró. *FreeLing*. URL https://http://nlp.lsi.upc.edu/freeling/index.php.

[25] and German Rigau Rodrigo gerri, Josu Bermudez. *Ixapipeline: Efficient and ready to use multilingual nlp tools*, 2014.

[26] spaCy GmbH. *paCy: Industrial-strength Natural Language Processing*, 2016. URL https://spacy.io/.

[27] Ewan Klein Steven Bird y Edward Loper. *Natural language processing with Python.*, 2009.

[28] W3schools. *PHP Basics*. URL http://www.w3schools.com/PHP/.

Este documento esta firmado por

Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Mon Jun 18 19:23:21 CEST 2018
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)