Aproximaciones a la simplificación léxica mediante aprendizaje profundo

Trabajo Fin de Máster

Juan Sixto Cesteros

Trabajo de investigación para el

Máster Universitario en Tecnologías del Lenguaje

Universidad Nacional de Educación a Distancia

Dirigido por la

Dra. Ana García Serrano

Septiembre 2023

Agradecimientos

Gracias a Ana por ser mi directora, por guiarme durante todo el trabajo y por respetar mis opiniones a lo largo del mismo. Gracias a Paula por todo, pero especialmente por cargar con cosas que no le tocaban para que yo pudiese sacar tiempo para esto. Y gracias a mis antiguos compañeros del MORElab Research Group, que también tienen parte de culpa en esto.

Resumen

Este trabajo aborda el problema de la simplificación automática de textos en español, con el propósito de transformar documentos de texto en versiones más accesibles, que faciliten su comprensión por diversos tipos de usuarios. Esta tarea implica retos tanto técnicos como conceptuales, con un potencial significativo para el bien social a través del acceso de muchas personas a la información. En este contexto, el presente trabajo aborda la tarea de simplificación centrándose en dos etapas específicas de la misma, la selección de términos sustitutos y la clasificación de los mismos, utilizando para ello enfoques basados en deep learning para la generación de soluciones múltiples de forma efectiva y versátil. La investigación se desarrolla en el ámbito de los textos en español, afrontando la tarea de generación de sustitutos a través de las herramientas más recientes en el ámbito del deep learning. Para lograrlo, se analizan y exploran los últimos avances y herramientas disponibles en el estado del arte. Luego, se llevan a cabo experimentos utilizando un conjunto de datos de referencia, que permite evaluar el rendimiento de los mismos con otros enfoques previamente publicados. La propuesta se basa en una aproximación a la tarea a partir de los trabajos de Aumiller y Gertz(Aumiller y Gertz, 2023), que obtienen los mejores resultados para el español en el marco de la tarea TSAR, incorporando los últimos modelos disponibles y explorando nuevas opciones de parametrización e ingeniería de instrucciones (*prompt engineering*).

El trabajo concluye con un análisis de los resultados obtenidos y el futuro de las tecnologías empleadas. Se discuten las fortalezas identificadas en las soluciones propuestas y las debilidades encontradas. Además, se abordan posibles áreas de mejora para investigaciones futuras, proporcionando una visión de la dirección que podrían tomar las tecnologías de simplificación automática de textos.

Abstract

This work addresses the problem of automatic text simplification in Spanish, with the purpose of transforming text documents into more accessible versions that facilitate their understanding by various types of users. This task involves both technical and conceptual challenges, with significant potential for social good through the access of many people to information. In this context, this paper addresses the simplification task by focusing on two specific steps of the task, substitute selection and substitute classification, using deep learning based approaches for generating multiple solutions in an effective and versatile way. The research is developed in the field of Spanish texts, facing the task of substitute generation through the most recent tools in the field of deep learning. To achieve this, the latest advances and tools available in the state of the art are analyzed and explored. Then, experiments are carried out using a benchmark dataset, which allows evaluating their performance with other previously published approaches. The proposal is based from the work of Aumiller and Gertz(Aumiller y Gertz, 2023), who obtain the best results for Spanish in the TSAR task framework, incorporating the latest available models and exploring new options for parameterization and instruction engineering (*prompt engineering*).

The work concludes with an analysis of the results obtained and the future of the technologies employed. The strengths identified in the proposed solutions and the weaknesses found are discussed. In addition, possible areas of improvement for future research are addressed, providing a vision of the direction that automatic text simplification technologies could take.

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Motivación

La motivación inicial del presente trabajo es el estudio de las aproximaciones basadas en deep learning para la tarea de simplificación léxica de textos. Una parte clave dentro del ámbito de la simplificación automática de textos, la sub-tarea de la simplificación léxica de textos tiene el objetivo de reemplazar las palabras complejas de un texto con sinónimos más sencillos de entender, preservando la información original del texto lo más fielmente posible. Además, sabemos que una de las características más relevantes durante los procesos de lectura es, presumiblemente, la disponibilidad de palabras más desarrolladas y completas (Anderson, Freebody, y others, 1981).

Recientemente, las tecnologías del lenguaje natural (PLN) han recibido importantes mejoras provocadas por los avances en las técnicas de aprendizaje profundo, en particular con la introducción de los grandes modelos lingüísticos (LLM) y el aprendizaje a través de *prompts*. Estos avances también en los artículos publicados durante los últimos años sobre simplificación léxica y sus subtareas, donde el uso de los grandes modelos (LLM) y los modelos enmascarados de lenguaje (MLM) han incrementado su popularidad, han pasado a ocupar las primeras posiciones en cuanto a rendimiento en el estado del arte.

Por un lado, la finalidad del proceso de simplificación léxica es facilitar la comprensión lectora a distintos tipos de usuario como estudiantes de lenguas extranjeras, lectores con discapacidades cognitivas como la dislexia, o con bajos niveles de alfabetización. Esto implica un elevado valor social a

su investigación, al tratarse de una tecnología de asistencia con gran potencial para facilitar el acceso de la gente a la información, factor clave para la autonomía personal y la inclusión social. Pero para lograrlo, es necesario desarrollar sistemas de simplificación automática altamente modulares, que permitan una fácil personalización de los componentes según las necesidades de simplificación de grupos particulares de usuarios (Štajner, 2021). Pero además, este proceso también se ha demostrado útil como herramienta PLN para pre-procesamiento de textos en otras tareas, como la traducción automática de textos o la generación de resúmenes.

Por todo esto, el presente trabajo aborda el problema general de la simplificación de textos a través de la sustitución de palabras en el ámbito de los textos en español, enfocado desde la perspectiva de los últimos modelos de aprendizaje profundo. También se plantea una discusión sobre las particularidades del aprendizaje basado en instrucciones (*prompts*) y sus implicaciones para trabajos futuros.

1.2. Objetivos específicos

Los objetivos específicos del trabajo que aquí se presenta son los siguientes.

OBJETIVO 1: Estado del arte sobre la simplificación automática de textos en español.

Para tener una visión general del conocimiento existente sobre la simplificación automática de textos, es necesario desarrollar un estado del arte. De esta forma, podemos examinar la literatura disponible así como los avances y tendencias de los estudios realizados hasta la actualidad. También se incluyen aquí los recursos, conjunto de datos y métricas existentes en el dominio específico.

A través de este proceso es posible obtener un punto de partida desde el que desarrollar nuestra investigación, así como un conocimiento crítico basado en la revisión y análisis de distintos tipos de textos. Finalmente, este objetivo también incluye una breve discusión sobre los descubrimientos, fortalezas y limitaciones descubiertos durante el proceso.

OBJETIVO 2: Análisis exhaustivo sobre las aproximaciones basadas en deep learning.

Dentro de la simplificación automática de textos, y una vez estableci-

dos los límites de la misma, se realiza un análisis en profundidad de las diferentes propuestas, tanto técnicas como teóricas, publicadas recientemente. En este sentido, destacan las recientes aproximaciones basadas en deep learning, y específicamente las que utilizan el aprendizaje basado en instrucciones (*prompts*), cuyo rendimiento lidera la tarea. Se realizará una contextualización histórica de las mismas, de los problemas y oportunidades que presentan, así como las herramientas más utilizadas.

OBJETIVO 3: Establecer un marco teórico y práctico de experimentación basado en la tarea TSAR de simplificación semántica.

Una vez analizados los límites de la tarea, se definirá un marco que permita llevar a cabo los diferentes experimentos planteados. De la misma forma, se identificarán las métricas adecuadas para realizar una evaluación de rendimiento a los modelos utilizados que permitan abarcar las características más relevantes. Para llevar a cabo esto se partirá de la base de la tarea compartida TSAR (*Shared Task on Multilingual Lexical Simplification* (Saggion et al., 2023)), extensamente utilizada en el estado del arte actual.

OBJETIVO 4: Experimentación con un modelo de deep learning en el ámbito específico de la tarea.

Se realizan distintos experimentos dentro del dominio específico de la tarea TSAR, que permite realizar comprobaciones con diferentes técnicas y parámetros, así como comparar los rendimientos obtenidos con otras aproximaciones existentes. Se experimenta con distintas opciones para la aplicación del modelo y la generación de instrucciones, y cómo estos cambios afectan al rendimiento del modelo en sus diferentes métricas. Finalmente, se extraen conclusiones sobre los resultados obtenidos y se tratan de aplicar a los problemas y oportunidades existentes hasta ahora.

1.3. Estructura del documento

A lo largo del capítulo 2 se exponen los conceptos básicos fundamentales vinculados con la tarea de simplificación automática de textos y las diferentes etapas en las que se divide la tarea. Se analizan las aproximaciones más relevantes existentes en el estado del arte y se realiza una revisión exhaustiva de los conjuntos de datos más utilizados en el ámbito académico, así como un listado detallado de las diferentes medidas de evaluación utilizadas frecuentemente.

El capítulo 3 detalla el caso de estudio, centrado en la generación de sustitutos de palabras complejas en español para múltiples candidatos a través de la tarea compartida TSAR-2022[1]. En este capítulo también se abordan las problemáticas inherentes al caso de estudio, se exploran las soluciones publicadas previamente y se presenta la propuesta de experimentación.

El capítulo 4 explica en detalle los experimentos realizados, así como la metodología utilizada para su desarrollo y evolución. También se detalla la solución técnica realizada y las diferentes técnicas de tratamiento de datos utilizadas. Además, se presentan los resultados obtenidos durante los mismos.

Finalmente, el capítulo 5 incluye una discusión de los resultados obtenidos y las dificultades encontradas durante el proceso. También se proporcionan las conclusiones derivadas del trabajo realizado, destacando las contribuciones de la investigación y se identifican las propuestas de trabajo futuro, así como las tendencias actuales en dicho campo.

[1] https://taln.upf.edu/pages/tsar2022-st/

Capítulo 2

Estado del arte

La tarea de simplificación automática de textos consiste, en términos generales, en transformar documentos de texto en otros más sencillos con el objetivo de facilitar su comprensión por diferentes usuarios finales, así como otras tareas del ámbito del PLN. A día de hoy se utiliza para beneficiar a usuarios con distintas dificultades para la lectura, como la afasia (Carroll et al., 1998) o la dislexia (Rello et al., 2013). Como consecuencia, existe una amplia variedad de aproximaciones técnicas a la simplificación automática de textos, también debido a la subjetividad implícita en el concepto de «simplificación» y la falta de consenso respecto a varias áreas del mismo (Grabar y Saggion, 2022). No obstante, existen dentro de la tarea varios procesos comunes a la mayoría de aproximaciones que permite comparar y mejorar en el desempeño general que las aplicaciones tienen en esta tarea.

En este capítulo se introduce la tarea de simplificación automática de textos y se desarrolla un contexto sobre las diferentes tecnologías y aproximaciones presentes en el estado del arte, así como los antecedentes históricos de los mismos. De forma complementaria, se definen los conceptos básicos utilizados en el resto del documento, aportando ejemplos aclaratorios. A continuación se definen los diferentes niveles de simplificación existentes en el estado del arte, dividiendo las técnicas más relevantes del estado del arte y desarrollando una estructura en etapas que abarquen el proceso de simplificación al completo. Una vez explicado esto, se desarrolla una revisión de los catálogos de datos más relevantes para tareas de simplificación y también de las distintas métricas de evaluación propuestas por diferentes autores para evaluar el rendimiento de la tarea. Finalmente, se elaboran las conclusiones y se desarrollan líneas de trabajo futuro.

2.1. Propósito de la simplificación automática de textos y contexto actual

La simplificación automática de textos está formada por un conjunto de métodos y técnicas capaces de adaptar un contenido textual con el propósito de facilitar el acceso a la información considerada relevante en un contexto particular. De esta forma se permite al usuario objetivo una mejor comprensión del texto, normalmente a costa de descartar elementos menos relevantes, como por ejemplo el estilo del autor o los matices propios de un léxico avanzado, pero que podrían resultar confusos. Debido a la especificidad de la tarea, no son frecuentes los planteamientos de carácter general, sino que existen multitud de propuestas enfocadas hacia diferentes grupos de usuarios finales, cada uno con sus propias características. También gracias a su gran potencial para el bien social y para mejorar la inclusión social de muchas personas, la simplificación léxica ha atraído cada vez más la atención de la comunidad de la PLN (Štajner, 2021). Entre las diferentes dificultades que intentan superarse mediante el uso de la simplificación automática destacan:

- Usuarios sin **conocimiento experto** de un ámbito concreto (Aluísio y Gasperin, 2010), como pueden ser los textos legales, médicos o históricos, donde se utilice un vocabulario específico del dominio.

- Usuarios con **discapacidades cognitivas** (Chen et al., 2017) que implican niveles bajos de lectura o con limitaciones en la alfabetización y la comprensión lectora, pero cuyo acceso a la información es un elemento clave en su toma de decisiones.

- **Usuarios infantiles** de distintas edades cuya capacidad de comprensión lectora se encuentra aún en desarrollo (Vu, Tran, y Pham, 2014).

- Usuarios con **afasia**, un trastorno del lenguaje que suelen sufrir las personas como consecuencia de un ictus o un traumatismo craneal y que suele provocar problemas para comprender los textos escritos (Carroll et al., 1998).

- Usuarios con **dislexia**, un trastorno del aprendizaje de la lectoescritura que se manifiesta en forma de dificultades en la lectura, como pueden ser omisiones, distorsiones, problemas de seguimiento visual y déficit

en la comprensión. Se sabe que el uso de determinadas condiciones textuales puede ayudar a paliar sus efectos (Rello et al., 2013).

- Usuarios con **trastornos del espectro autista (TEA)**, un grupo de discapacidades del desarrollo que pueden provocar problemas comunicacionales significativos entre otros (Evans, Orasan, y Dornescu, 2014). Esto puede implicar dificultades para inferir información contextual o comprender verbos mentales o lenguaje emocional, así como frases largas con estructuras sintácticas complejas (Tager-Flusberg, 1981).

- Usuarios de una **segunda lengua** (Paetzold, 2016), que pese a comprender el idioma pueden tener dificultades a la hora de afrontar estructuras semánticas complejas, vocabularios atípicos o figuras literarias.

Además, la tarea de simplificación automática de textos también tiene un papel destacado como tecnología de ámbito social, pues la CRPD (ONU, 2006) de la ONU establece que el derecho a una información accesible es un derecho fundamental que debe concederse a todas las personas. Este es un factor clave para la capacitación personal y la inclusión social y, sin embargo, la información textual que se encuentra en la web, en las noticias y en otras fuentes suele ser lingüísticamente demasiado compleja para muchas personas, lo que impide su participación activa en la sociedad.(Štajner, 2021).

De forma similar, las técnicas de simplificación automática de textos son también útiles a la hora de facilitar información relevante a otros sistemas NLP , ejerciendo como un primer filtrado en el contenido textual de herramientas más complejas (Al-Thanyyan y Azmi, 2021). En el estado del arte actual se pueden encontrar muchas aproximaciones a distintas tareas de procesamiento de texto natural que incluye la simplificación automática como un elemento más dentro del conjunto de técnicas de pre-procesamiento. No obstante, es preciso señalar que estas técnicas de simplificación, al igual que ocurre con las tareas generales, suelen enfocarse de formas muy distintas entre ellas, especialmente en lo relativo a la evaluación de sus sistemas. De hecho, una ventaja del uso de simplificación como parte de una tarea más compleja es que la mejora de rendimiento que aporta su uso puede medirse mediante métricas más claras, en comparación con las tareas de simplifica-

ción pura. Entre las distintas tareas NLP en las que se ha probado la utilidad de las técnicas de simplificación automática destacamos las siguientes:

- **Parsing**, donde la simplificación utilizada como filtro de pre-procesamiento mejora el análisis de las cadenas de caracteres del lenguaje natural conforme a las reglas de la gramática formal (Oliveira, Wong, y Hong, 2010).
- **Summarization** donde una correcta simplificación permite capturar de forma más sencilla el contenido importante en el menor espacio posible (Vanderwende et al., 2007).
- **Information extraction**, a través del preprocesamiento de frases complejas a otras más sencillas, especialmente en las relaciones de subordinación y coordinación, que implican unidades sintácticas del mismo rango en una frase (Beigman Klebanov, Knight, y Marcu, 2004) (Evans, 2011). También puede aplicarse a la extracción de información en dominios específicos, como por ejemplo en textos financieros (Kang et al., 2022b), que implican complejidades acionales.
- **Machine translation**, en la cual el vocabulario utilizado y la longitud de las frases son un elemento clave en su desempeño (Štajner y Popović, 2016), (Hasler et al., 2017).

Según Siddhartan (Siddharthan, 2014), la finalidad de la tarea de simplificación de textos es reducir la complejidad léxica y sintáctica de un texto y, al mismo tiempo, intentar preservar su originalidad haciendo uso de frases más cortas, estructuras más simples y vocabulario más sencillo. Esto es, hacer que la información sea más comprensible y accesible.

La simplificación léxica con herramientas automatizadas es una tarea necesaria debido a la costosa y lenta transformación manual de un lenguaje complejo a uno básico, lo que requiere editores profesionales con un extenso conocimiento del idioma. Además, la tarea puede enfocarse en diferentes grupos de usuarios, como por ejemplo hablantes nativos y no nativos que requieren simplificaciones diferentes según su vocabulario (Yimam et al., 2017). De la misma forma, personas con limitaciones cognitivas o de lectura tienen necesidades aún más variadas de simplificación del vocabulario, al igual que los estudiantes de idiomas también requieren una simplificación personalizada que depende de su nivel de habilidad lingüística y lengua

materna. Por todo esto, la simplificación automatizada de textos es una herramienta muy potente para reducir costos y mejorar la comprensibilidad de un público más amplio, así como personalizar un mismo texto para distintos usuarios.

2.2. Conceptos básicos de simplificación de textos

En esta sección se describen los conceptos clave del estado actual de la técnica de simplificación de textos. Esta terminología se mantendrá durante el resto del documento para facilitar su comprensión. Además, en algunos casos se han unificado términos que se nombran de varias formas en el estado del arte para evitar confusiones.

La **Simplificación Automática de Textos**, simplificado como **ATS** (*Automatic Text Simplification*) abarca cualquier proceso que transforme un texto de forma automática, mediante técnicas y algoritmos aplicados al lenguaje natural. En caso contrario, se habla de **Simplificación Manual de Textos**, o **MTS** (*Manual Text Simplification*) de aquí en adelante, haciendo referencia a cualquier desarrollo de simplificación que utilice usuarios humanos para modificar el texto.

Definimos como **Usuario Final** (*End User*) a la persona o personas que van a utilizar de forma directa una herramienta de Simplificación Automática de Textos. Para realizar una evaluación correcta de la tarea de simplificación, ésta debe depender del usuario final y tener en cuenta sus necesidades específicas. Tal y como indica (Grabar y Saggion, 2022), los distintos usuarios pueden necesitar estrategias de simplificación diferentes, al igual que los distintos tipos de documentos y su contenido difieren, y en consecuencia la adaptación necesaria de dichos documentos. Además, el nivel de alfabetización de las personas puede variar mucho, incluso dentro de una misma población. Para referirnos a grupos amplios de usuarios con problemáticas comunes usaremos el término **población específica de lectores**.

Un **corpus** o **conjunto de datos** puede definirse como una colección de textos naturales, legibles por máquina (incluidas transcripciones de datos hablados) que se muestrea para que sea representativa de una lengua natural o variedad lingüística concreta (McEnery, Xiao, y Tono, 2006). Para añadir algo de profundidad a esta definición, (Rojo, 2016) aclara que los textos deben ser naturales, en lugar de artificiales o creados expresamente para su

incorporación al corpus, deben almacenarse en formato electrónico para poder recuperar la información que precisemos de forma automática (legibles por máquina), tienen que ser representativos de la variedad de la que proceden y deben permitir su estudio científico (no exclusivamente lingüístico), lo cual suele implicar la adición de información gramatical, léxica y pragmática al texto.

Los corpus desempeñan un papel esencial en la investigación del procesamiento del lenguaje natural (PLN) y en otras investigaciones lingüísticas, y a su vez el desarrollo de estas nuevas investigaciones permite crear corpus cada vez más extensos y útiles. Esto ocurre también en el ámbito específico de la simplificación automática de textos, donde también son llamados **textos de referencia** *Reference data* en muchas publicaciones. Tal y cómo explica (Grabar y Saggion, 2022), los datos de referencia desempeñan un papel muy importante en el desarrollo y la evaluación de los sistemas de simplificación, pudiendo encontrar diferentes formas de enfrentar su creación en el estado del arte actual, con sus ventajas y limitaciones.

En la ámbito de las tareas de procesamiento del lenguaje natural se habla del **dominio específico** del corpus para referirse al ámbito que tienen en común todos los documentos pertenecientes al mismo y que implican una serie de reglas y observaciones aplicables a todos ellos. Algunos dominios extensamente utilizados en la literatura para la tarea de simplificación son los documentos médicos (Cardon y Grabar, 2020), así como los técnicos (Ermakova et al., 2021) y los de relativos a textos legales (Cemri, Çukur, y Koç, 2022). Como bien explica (Grabar y Saggion, 2022), el dominio de los documentos de un corpus influye en el resultado del proceso de simplificación y también en la evaluación del mismo. Los documentos de lenguajes generales y especializados requieren aproximaciones técnicas distintas, así como recursos léxicos diferentes. En el caso de los documentos pertenecientes a dominios especializados, estos pueden incluir una terminología específica que requiera una simplificación intensa a nivel léxico y semántico o poseer estructuras sintácticas propias que exijan su simplificación sintáctica. Cuando un corpus no tiene un dominio concreto se denomina como ≪de dominio general≫.

Tal y cómo indicaba (Siddharthan, 2014), en la actualidad no existe un consenso sobre la metodología a aplicar para evaluar la calidad de un proceso o técnica de simplificación de textos. Aunque este tema se trata en

detalle más adelante, sí existen dos grupos claramente divididos de métricas de evaluación; la **evaluación humana** (*human evaluation*), basadas en el juicio humano como pilar fundamental para determinar el correcto funcionamiento de la simplificación, y la **evaluación automática** (*automatic evaluation*), más enfocada en el uso de métricas objetivas y la similaridad respecto al texto original. Aquí también es importante reseñar el concepto de **legibilidad** (*readability*), una métrica ampliamente utilizada para representar la facilidad de un texto para ser comprendido por el lector y para la que se han desarrollado un gran número de fórmulas (Crossley, Allen, y McNamara, 2011).

2.3. Niveles de simplificación

La simplificación de textos se puede afrontar de diversas formas, dependiendo del objetivo final y de la definición particular del proceso de simplificación. Por ejemplo, en muchas ocasiones se considera el proceso de simplificación como el intercambio de las palabras más complejas de comprender que componen un texto por otras más sencillas, accesibles o adaptadas al público objetivo. Sin embargo, este problema también se puede afrontar descomponiendo las frases más complejas en otras más sencillas, o combinar ambos planteamientos. A su vez, estas aproximaciones pueden dividirse en distintas sub-tareas o niveles, que frecuentemente se investigan por separado, centrándose en uno o más niveles que puedan trabajar en conjunto.

Al llevar a cabo la tarea de simplificación es importante conocer los diferentes niveles a los que puede modificarse un texto y las características particulares de cada uno. Por ejemplo, los primeros sistemas de simplificación automática funcionaban únicamente en base a reglas preestablecidas donde la simplificación sintáctica era clave, estando diseñados como subsistemas integrados en herramientas más complejas dedicadas a otras tareas de PLN. Sin embargo, con el avance del tiempo y la llegada de nuevas técnicas, los sistemas de simplificación automática de textos han crecido en complejidad y especificidad, integrando diferentes operaciones de simplificación, como la simplificación léxica (simplificación de palabras), la simplificación sintáctica o la generación de explicaciones (Štajner, 2021).

Pero a pesar de que muchos de los desarrollos en estas tareas se han realizado históricamente por separado, todas ellas se encuentran fuertemente

relacionadas y deben tenerse en cuenta a la hora de mejorar su desempeño. Por ejemplo, aunque la simplificación léxica puede abordarse como una simple sustitución de palabras por otras más sencillas, durante una simplificación sintáctica se realizan cambios en la estructura de una frase, por ejemplo dividiéndola en otras más sencillas, a través de cambios que pueden implicar sustituciones de palabras para mantener la coherencia. De la misma forma, algunos cambios a bajo nivel pueden ser más eficaces a través de cambios en el contexto global, de la misma forma que algunos cambios en la estructura de las frases pueden implicar correcciones a nivel de palabra, como por ejemplo, en el género o el número de los adjetivos.

Lógicamente, muchos sistemas actuales utilizan una combinación de enfoques para abordar la tarea de simplificación de textos a través de múltiples metodologías centradas en los diferentes aspectos de los textos. A continuación se explican los distintos niveles existentes en el estado del arte, clasificados en cuatro categorías; simplificación léxica, simplificación sintáctica, generación de explicaciones y la traducción automática.

2.3.1. Simplificación léxica

La simplificación léxica puede definirse como una tarea del procesamiento del lenguaje natural (PLN) cuyo objetivo es simplificar automáticamente las palabras de un texto u oración para que la información sea más comprensible para el usuario final. El nivel de simplificación aplicado afecta directamente a la cantidad de información que se pierde respecto del significado original, aunque la tarea de simplificación debe siempre preservar el significado original del texto, eliminando únicamente la información no esencial. Las aproximaciones de simplificación léxica suelen utilizar recursos lingüísticos para encontrar sinónimos, en combinación con diferentes algoritmos para detectar que palabras tienen un elevado índice de complejidad que permitan detectar cuales cambiar.

Para facilitar la comprensión de los procedimientos involucrados en la tarea de simplificación léxica para los diferentes enfoques en la literatura, durante este trabajo utilizamos la definición de Shardlow (Shardlow, 2014) de las distintas subtareas. Estas permiten separar la tarea en cinco pasos consecutivos que pueden afrontarse por separado, pudiendo medir sus rendimientos en cada etapa de cara a comparar con otras aproximaciones y optimizar cada paso. Según esta definición, la simplificación léxica pue-

de dividirse en cinco subtareas que pueden modelizarse por separado, de forma modular, o conjuntamente: (1) identificación de palabras complejas (*Complex Word Identification* (CWI)); (2) generación de posibles sustitutos (*Substitute Generation* (SG)); (3) selección de sustitutos que se ajusten al contexto y preserven el significado original (*Substitute Selection* (SS)); y (4) clasificación de sustitutos (*Substitute Ranking* (SR)).

1. Complex Word Identification (CWI)
2. Substitute Generation (SG)
3. Substitute Selection (SS)
4. Substitute Ranking (SR)
5. Morphological generation and context adaptation

Complex Word Identification (CWI)

El objetivo de la primera subtarea CWI es diferenciar aquellas palabras que deben simplificarse de las que no para asegurar que sólo se modifican aquellas que resulten difíciles para el usuario final, respetando las que pueden comprenderse con facilidad y se respeta en la máxima medida el contenido original.

No obstante, algunos sistemas de simplificación léxica presentes en la literatura no llevan a cabo la tarea de identificar palabras complejas antes de tratar de simplificarlas. En su lugar, estas aproximaciones consideran todas las palabras del texto como potencialmente difíciles y utilizan las demás subtareas para tratar de encontrar una sustitución adecuada. Sin embargo, algunos autores como (Paetzold y Specia, 2015) han demostrado que agregar un módulo de identificación de palabras complejas al principio del proceso de simplificación léxica puede mejorar el rendimiento al evitar tratar de simplificar palabras de forma innecesaria.

Además de su integración en sistemas que afrontan el proceso de simplificación al completo, la tarea de identificación de palabras complejas también se ha afrontado de forma independiente en la literatura existente. De esa forma se permite avanzar en las diferentes técnicas de identificación y comparando sus rendimientos, mejorando de esta forma el rendimiento general de los sistemas en el futuro. En esta tarea destacan dos tareas compartidas; SemEval 2016 CWI para inglés (Paetzold y Specia, 2016b) y BEA 2018

CWI para inglés, alemán y español. Aquí cabe destacar la tarea compartida SemEval 2021 sobre predicción de complejidad léxica (Shardlow et al., 2021), donde sus autores desarrollan un nuevo conjunto de datos centrado en anotaciones continuas de tres dominios distintos sobre identificación de palabras complejas.

Substitute Generation (SG)

El objetivo de esta etapa del proceso de simplificación es la generación de una lista de palabras candidatas a utilizarse como sustituto de cada una de las palabras complejas identificadas previamente. Estas palabras candidatas suelen obtenerse a partir de diccionarios especializados o grandes corpus de textos pertenecientes al dominio adecuado. Lógicamente, este proceso influye de forma directa en el resto de etapas y en el rendimiento final del proceso de simplificación, y es una de las partes más utilizadas y estudiadas en la literatura. La mayor dificultad durante esta etapa es evitar que se produzcan demasiados candidatos falsos que compliquen la toma de decisión de los modelos empleados en los pasos siguientes. En la literatura encontramos fundamentalmente dos enfoques existentes para la generación de substitutos: Bases de datos lingüísticas (*Linguistic Database Querying*) y Generación automática (*Automatic Substitution Generation*) (Paetzold y Specia, 2017b).

Las **Bases de datos lingüísticas** son recursos construidos de forma manual por agentes expertos basándose en la experiencia y el conocimiento humano a la hora de sustituir palabras por otras más sencillas. Como conveniente principal encontramos la dificultad intrínseca, tanto en términos de tiempo como económicos, para crear estos recursos con la amplitud suficiente como para cubrir todos los casos existentes. además de su dependencia del idioma. Estos problemas suelen reducirse en los casos donde la simplificación automática se aplica a dominios técnicos específicos, como pueden ser los términos médicos (Kandula, Curtis, y Zeng-Treitler, 2010). En el caso de ámbitos de carácter general, lo más frecuente es utilizar bases de datos léxicas generales, siendo la más destacada WordNet (Miller, 1995), de la que pueden extraerse sinónimos de forma automática. Durante los últimos años muchas aproximaciones utilizan ambos tipos de bases de datos de forma coordinada, pata tratar de abarcar más terreno. En el dominio del español podemos destacar LexSiS (Bott et al., 2012) que utiliza un modelo de vecto-

res de palabras para encontrar posibles sustitutos de una palabra objetivo, junto a un procedimiento de cálculo de simplicidad basado en un estudio de corpus e implementado en función de la longitud y la frecuencia de las palabras. Para ello, utiliza el tesauro OpenThesaurus (Naber, 2004) junto a un corpus de documentos en español procedentes de la Web, obteniendo resultados con buenos porcentajes de preservación del significado al ofrecer mayor número de sustituciones léxicas respecto a otros sistemas similares.

En cambio, las técnicas de **generación automática** tratan de generar candidatos a partir de otros recursos menos costosos que los diccionarios especializados, como por ejemplo, diccionarios básicos que incluyen descripciones de palabras. En los modelos desarrollados con este enfoque podemos encontrar dos recursos especialmente utilizados; Wikipedia y WordNet. En el caso de Wikipedia y Simple Wikipedia para el inglés, la simplificación de textos se realiza generalmente a través de la extracción de simplificaciones léxicas y sintácticas a partir de frases y sus versiones simplificadas. Por otra parte, la base léxica Wordnet cuenta con relaciones de sinonimia e hiperonimia que pueden adaptarse a modelos contextuales que permiten extraer relaciones entre palabras. También se ha demostrado que la combinación de ambos recursos acostumbra a mejorar los resultados (Paetzold y Specia, 2017a).

Substitute Selection (SS)

Durante esta tarea los sistemas deben decidir cuales de las sustituciones candidatas generadas en la fase previa puede reemplazar a la palabra o palabras complejas sin comprometer la gramaticalidad, el contexto o el significado de la frase. No se trata de encontrar al candidato idóneo, sino de realizar un primer filtrado para evitar que palabras que no son sinónimos reales de la palabra compleja puedan ser seleccionadas. Esto suele fundamentarse en ajustar los posibles candidatos al contexto de la frase que se está simplificando , evaluando su construcción gramatical y su significado. Algunos autores, como (Paetzold y Specia, 2017b) consideran esta tarea como «una de las más importantes del proceso de simplificación léxica, ya que debe evitar que un sistema realice sustituciones léxicas que alteren el significado o la fluidez de una frase compleja, lo que, en algunos casos, la haría incomprensible».

Para afrontar este problema existe en el estado del arte diferentes técni-

cas consolidadas, entre las que destacan el Explicit/Implicit Sense Labelling, el Semantic Similarity Filtering y el Part-of-Speech Tag Filtering. A continuación se describen brevemente las tres técnicas y las implicaciones que tienen en el resto del sistema de simplificación.

Explicit/Implicit Sense Labelling (Etiquetado de significado implícito/explícito): Estos enfoques técnicos de simplificación léxica exploran la selección de sustitutos utilizando métodos de clasificación para decidir la etiqueta de significado de una palabra objetivo ambigua dentro del contexto de la frase que se está simplificando y, a continuación, selecciona como candidatas válidas las que tienen la misma etiqueta. En el caso de los etiquetados explícitos, para obtener estas etiquetas de significado suelen utilizarse los códigos synset de WordNet(Miller, 1995) en la mayoría de aproximaciones presentes en el estado del arte, como por ejemplo (Nunes et al., 2013) o (Navigli y Ponzetto, 2010), donde la evaluación se realiza midiendo la distancia semántica entre la frase/contexto original y su versión simplificada. Otro enfoque en esta subtarea es el de (Baeza-Yates, Rello, y Dembowski, 2015), consistente en extraer el 5-grama alrededor de la palabra compleja a simplificar y utilizarlo para obtener puntuaciones de sus posibles sinónimos a partir de la frecuencia del nuevo 5-grama en el corpus de Google 1T (Brants y Franz, 2006). De esta forma, aquellos con una frecuencia más alta son valorados como candidatos más adecuados.

Para tratar de mejorar las técnicas explícitas, las aproximaciones implícitas de etiquetado usan métodos automáticos para aprender clasificaciones de significados de palabras complejas, en lugar de consultarlas en recursos externos. No obstante, apenas hay aproximaciones de este tipo que hayan demostrado resultados eficaces por la complejidad de la gestión de los datos a la hora de definir las diferentes clases de significados.

Semantic Similarity Filtering (filtrado de similitud semántica): El filtrado de similitud semántica consiste en establecer una métrica de la similitud entre el significado de una palabra compleja en su contexto y el de un sustituto candidato, y luego descartar todas las candidatas que no tengan suficiente similitud de significado con la palabra compleja (Paetzold y Specia, 2017b). Un ejemplo de esto es la biblioteca de software HESML (Lastra-Díaz, Lara-Clares, y Garcia-Serrano, 2022), diseñada dentro del ámbito biomédico y que utiliza un método de aproximación del algoritmo de Dijkstra para taxonomías que permite el cálculo en tiempo real de cualquier medida de

similitud semántica basada en rutas. Estos métodos utilizan un modelo de incrustación (*word embeddings*) de palabras que evalúa la similitud semántica entre un candidato y la frase/contexto de una palabra compleja, clasificando a los candidatos en función de una distancia preestablecida, como por ejemplo el coseno entre cada uno de ellos y las palabras de contenido de la frase.

Part-of-Speech Tag Filtering (Filtrado de etiquetas de parte de discurso): Como alternativa al significado de las palabras, algunos enfoques del estado del arte utilizan etiquetas POS como sustitutos de las etiquetas de significado, seleccionando como candidatas válidas aquellas palabras candidatas que tienen la misma etiqueta POS que la palabra compleja en su contexto. También se ha utilizado un concepto similar que combina las etiquetas POS con el aprendizaje automático de reglas de sustitución léxico-sintácticas a través de la transducción de árboles.

Sin embargo, aunque esta aproximación parece funcionar bien únicamente en el caso de palabras que poseen distintos roles gramaticales, no obtiene buenos resultados a la hora de distinguir los diferentes significados semánticos (Paetzold y Specia, 2013).

Substitute Ranking (SR)

La última subtarea habitualmente es la toma de decisiones sobre los sustitutos candidatos generados en los pasos anteriores en la frase/contexto. En muchos sistemas de simplificación léxica la finalidad de la tarea consiste en escoger un único candidato válido, aunque en algunos planteamientos, como en la tarea TSAR 2022 (Saggion et al., 2023), lo que se requiere del sistema es una lista de candidatos ordenados. Este segundo planteamiento puede adaptarse con más facilidad a distintos contextos de simplificación o a grupos de usuarios objetivo con características específicas. Pero en ambos planteamientos, la tarea de clasificación de sustituciones (SR) suele consistir en cuantificar la simplicidad de las sustituciones candidatas de modo que la sustitución de una palabra compleja por la mejor candidata produzca el resultado más sencillo posible. Las principales aproximaciones a esta subtarea se recogen en (Paetzold y Specia, 2017b) , que las divide en tres categorías principales, las cuales describimos a continuación.

Frequency-based (Basado en frecuencias): Las estrategias basadas en frecuencias son las más comunes en el estado del arte dentro del ámbito

de la simplificación léxica. Son aproximaciones sencillas, basadas en la idea de que las palabras que se utilizan con más frecuencia son más familiares para los lectores y, por tanto, más comprensibles. Una idea que ya hemos visto presente en gran parte de los desarrollos relativos a simplificación léxica, pero que tiene sus contraprestaciones, en especial cuando lo aplicamos a grupos de usuarios objetivo concretos o a dominios específicos, donde el lenguaje puede tener sus propias peculiaridades. Para estimar esta frecuencia se recurre normalmente a corpus amplios, normalmente de dominios generalistas, aunque también a motores de búsqueda, con resultados aceptables en muchos casos. De hecho, estos enfoques basados en la frecuencias han demostrado su eficacia en muchos casos, superando a otros enfoques de clasificación más sofisticados. Además, estos enfoques suelen ser extrapolables con facilidad a otros idiomas, y con resultados similares, especialmente en el caso de conjuntos de datos existentes ya en múltiples idiomas.

Simplicity measures (Métricas de simplicidad): Otro enfoque para evaluar los diferentes candidatos son las métricas de simplicidad, que tratan de combinar diferentes características de la frase/contexto y sus candidatos para representar la simplicidad de una palabra. En el estado del arte podemos encontrar métricas diferentes para realizar esta tarea, como pueden ser la definición de complejidad propuesta por (Biran, Brody, y Elhadad, 2011) , que utiliza la complejidad léxica y la complejidad de corpus para calcular su puntuación, o el índice de complejidad presentado por (Glavaš y Štajner, 2015), que se basa en la hipótesis de que la informatividad (*informativeness*) de una palabra muestra correlación con su complejidad. La principal ventaja de estos enfoques respecto a los basados en frecuencia es la capacidad de incorporar diferentes métricas a la puntuación total, de forma que se tengan en cuenta múltiples características de los candidatos y su contexto, que además pueden ponderarse hasta encontrar un equilibrio. Estos enfoques se han evaluado en distintas tareas compartidas, donde destacan varias ediciones de SemEval en la que se compararon numerosas métricas.

Machine learning-assisted (Asistidas por aprendizaje automático): Siguiendo la tendencia general en casi todos los ámbitos del PLN, las técnicas de aprendizaje automático también han ganado una fuerte presencia en esta subtarea en los últimos años. Estos hacen uso de diferentes clasificadores para obtener u ordenar los candidatos a través de diferentes parámetros, muchos de los cuales ya estaban presentes en los enfoques de

métricas de simplicidad. Aunque en el estado del arte podemos encontrar muchas aproximaciones que utilizan clasificadores basados en representación vectorial, como SVM, recientemente han aparecido otros basados en redes neurales, como (Paetzold y Specia, 2017a), que demuestran tener mejores resultados. Estos utilizan un perceptrón multicapa para determinar la clasificación entre los distintos candidatos a partir de datos anotados. Este enfoque es también una opción muy adecuada cuando no se dispone de grandes conjuntos de datos para entrenamiento sobre el que calcular las frecuencias y las métricas, puesto que pueden utilizarse otras características del texto, así como modelos de lenguaje.

Morphological generation and context adaptation

Esta última etapa no tiene lugar en todos los sistemas, pero es un paso necesario en aquellos que utilizan las formas simplificadas de las palabras durante el proceso y que finalmente deben adaptarse al contexto de la frase. Aquí se comprueban las características del sustituto respecto al contexto y se modifican para que su uso sea correcto, por ejemplo adaptando el género y el número de la palabra para ser coherente con el resto de la frase. También puede revisarse aspectos relativos a la corrección orto-tipográfica o a la construcción de la frase, en especial en las aproximaciones enfocadas a grupos de usuarios específicos.

2.3.2. Simplificación sintáctica

La simplificación sintáctica consiste en identificar las complejidades gramaticales de una frase/contexto objetivo y transformarlas en estructuras más sencillas que faciliten la comprensión del usuario. Esta reducción de la complejidad sintáctica se puede aplicar de muchas maneras y no existe una única estrategia definida para afrontarla. Para ello, (Shardlow, 2014) presenta dos técnicas fundamentales, estas consisten en dividir las frases largas en otras más pequeñas y reescribir las frases en forma pasiva a una activa. De esta forma, aunque se pueden perder matices en los textos, se facilita la asimilación de la información por parte de los lectores, en especial en casos con dificultades añadidas, como las explicadas en la sección 2.1.

La simplificación léxica está estrechamente relacionada con un sub-campo de la PLN específico, el de reescrituras de textos con distintos objetivos, de donde suelen exportase técnicas basadas en reglas y modelos lingüísticos. De

estas, las principales técnicas basadas en reglas para simplificar se pueden agrupar en los grupos que se describen a continuación.

División de frases: Estas técnicas se basan principalmente en dividir frases complejas en otras más sencillas sin dañar el significado original y manteniendo su gramaticalidad(Collados, 2013). Este proceso de simplificación da lugar a un conjunto de frases nuevas que deben ordenarse correctamente para comprender plenamente el mensaje de la frase original, de lo contrario la simplificación no es efectiva, porque en algunos casos es necesario reordenar la frase respecto al orden original. Esta reordenación no es una tarea difícil para un nativo español, pero es mucho más difícil realizarlo en base a reglas. Por ello, recientemente han empezado a utilizarse sistemas de aprendizaje automático para abordar esta tarea.

Reordenación de oraciones: El proceso de reordenación (o reorganización) consiste en alterar el orden de las frases complejas para facilitar su lectura, aunque no implica que estas se hayan dividido previamente. Por ejemplo, estas técnicas suelen utilizarse para simplificar la información en frases subordinadas (Gasperin, Maziero, y Aluisio, 2010), que no suelen poder dividirse sin perder su significado original.

Eliminación de oraciones: Para simplificar los textos, este planteamiento utiliza diversas operaciones de simplificación a nivel de frase, fundamentalmente dividir, eliminar y reducir textos, para generar versiones más sencillas de un texto. Estas operaciones pueden entrenarse mediante modelos de lenguaje a partir de conjuntos de datos simplificados. Un ejemplo de esto es el estudio de (Štajner, Drndarevic, y Saggion, 2013), que aborda el problema de simplificación automática de textos en español con el fin de hacerlos más accesible a las personas con discapacidades cognitivas.

2.3.3. Generación de explicaciones

Otra forma de simplificación de texto para facilitar su comprensión es la generación de explicaciones adicionales. Se trata de un conjunto de técnicas destinadas a identificar conceptos difíciles en un texto y enriquecerlos con información adicional para mejorar la comprensión del usuario y su contextualización. Esta generación de explicaciones tiene también un potencial relevante en ámbitos educativos o aquellos que suelen implicar grandes datos de información relacionada, como pueden ser documentos históricos. Además de la presentación de información adicional respecto a términos complejos,

la generación de explicaciones también puede utilizarse mediante sustituciones o adiciones del texto original. Por ejemplo, incluyendo frases sencillas que expliquen el término original de forma intercalada en el texto original.

Aun así, se trata de una técnica propensa a errores, que además pueden provocar engaños al usuario final, complicar aún más un texto o sencillamente resultar poco útil(Shardlow, 2014). Esto implica que los errores pueden penalizar más que en otros planteamientos, y que debe de tenerse en cuenta a la hora de ponderar la veracidad de las explicaciones. Lógicamente, estas técnicas también resultan útiles en combinación con otras técnicas de simplificación léxica o sintáctica

2.3.4. Traducción automática estadística

La última de las técnicas de simplificación es la traducción automática, una técnica que puede considerarse como otra tarea dentro del ámbito del procesamiento del lenguaje natural, pero que también tiene su espacio dentro de la simplificación de textos. Esta consiste en diferentes métodos estadísticos que transforman un texto de un idioma a otro mediante cambios léxicos y semánticos. En este sentido, dentro de la simplificación de textos, podemos considerar a nivel teórico la simplificación como un proceso de traducción desde un lenguaje complejo a su versión simplificada, como ocurre por ejemplo con los textos de Wikipedia y los de Simple Wikipedia, su versión destinada a niños y adultos que están aprendiendo el idioma [1]. Un ejemplo exitoso es el trabajo de (Nelken y Shieber, 2006), que alinea frases pertenecientes a corpus monolingües como primer paso de un entrenamiento para reescritura de textos, dentro del ámbito de la generación de resúmenes.

2.4. Medidas de evaluación

En el ámbito de las tareas de clasificación automática en el PLN, existen en la literatura varias métricas no equivalentes que permiten medir diferentes aspectos de acierto de un sistema. Cada métrica representa de una manera diferente el grado de acercamiento de la solución propuesta por el modelo respecto a la solución correcta. Las subtareas que forman la simplificación léxica se evalúan comúnmente mediante la exactitud, la precisión, la exhaustividad y el valor F, aunque también se han utilizado algunas métricas

[1] https://simple.wikipedia.org/

adicionales, como el potencial y el MAP (*Mean Average Precision*)(North et al., 2023).

A continuación se presentan las medidas básicas comúnmente utilizadas en la evaluación de corpus lingüísticos y en el estado del arte presentado previamente. Después, en el apartado 2.4.5 se presentan las métricas utilizadas para los sistemas de múltiples candidatos, que tienen su propia casuística.

2.4.1. Matriz de confusión

La **Matriz de confusión (*Confusion matrix*)** es una herramienta de de visualización de rendimiento de un algoritmo de aprendizaje en el contexto del aprendizaje supervisado categórico. Se trata de una tabla en dos dimensiones, valor real y valor propuesto por el algoritmo, en el que cada columna representa el número de instancias de cada clase. De esta manera, podemos observar la proporción de clasificaciones correctas e incorrectas para cada categoría de manera sencilla.

Si consideramos un sistema de clasificación de dos categorías como una evaluación binaria respecto a una única categoría, es decir, si un elemento pertenece o no a ella, podemos establecer una serie de métricas sencillas sobre el rendimiento del algoritmo a la hora de comprender dicha categoría. Por cada resultado de un sistema de evaluación binario podemos identificar cuatro resultados posibles.

- **Verdadero Positivo (true positive - TP):** cuando la conclusión del clasificador indica una categoría determinada y es correcto.
- **Verdadero Negativo (true negative - TN):** cuando la conclusión del clasificador indica que el elemento no pertenece a una categoría determinada y es correcto.
- **Falso Positivo (false positive - FP):** cuando la conclusión del clasificador indica una categoría determinada y es errónea.
- **Falso Negativo (false negative - FN):** cuando la conclusión indica que el elemento no pertenece a una categoría determinada y es erróneo.

2.4.2. Exactitud

La **Exactitud (*Accuracy*)** es una medida estadística que calcula en una prueba de clasificación binaria la cantidad de elementos identificados

correctamente. Se trata de la medida estadística más sencilla. En un sistema de clasificación multicategoría, la exactitud se mide como el número de aciertos del sistema entre la cantidad total de elementos clasificados.

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

La Paradoja de Exactitud

La **paradoja de exactitud (*accuracy paradox*)** establece que, en ocasiones, un modelo predictivo puede poseer un poder predictivo superior a otro con una puntuación de **Exactitud** mayor. Por esta razón, se consideran otras medidas como la **Precisión** y la **Exhaustividad** más efectivas a la hora de medir el éxito de un modelo predictivo.

La Exactitud es la medida mas básica para un modelo predictivo y parece lógico que a mayor número de aciertos más efectivo es un modelo predictivo. En efecto, la Exactitud es un criterio básico que nos da una idea del buen funcionamiento de las predicciones realizadas, sin embargo, existen casos en los que una Exactitud muy elevada puede pertenecer a un modelo predictivo unútil para la terea que realiza. Este problema tiene lugar especialmente cuando el conjunto de categorías del modelo se encuentra sensiblemente desbalanceado. En esos casos, un modelo predictivo puede predecir el valor de la clase mayoritaria en la totalidad de los casos, para obtener una puntuación alta en Exactitud. Sin embargo, ese modelo no es útil ya que no realiza la tarea para la que se ha desarrollado.

Para evitar este problema, numerosos investigadores de la literatura recomiendan (Powers, 2020) complementar la Exactitud con otras métricas como pueden ser la Precisión y la Exhaustividad.

2.4.3. Precisión y exhaustividad

La **Precisión (*Precision*)** se define numéricamente como la proporción de verdaderos positivos contra todos los resultados positivos (tanto verdaderos positivos, como falsos positivos). Muestra la proporción de resultados correctos respecto al todos los resultados asignados a esa categoría por el clasificador.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

La **Exhaustividad (*Recall*)** se define numéricamente como la proporción de los verdaderos positivos contra todos los elementos positivos del conjunto. Muestra la proporción de resultados correctos respecto a todos los elementos pertenecientes a esa categoría.

$$Exhaustividad = \frac{TP}{TP + FN} \tag{2.3}$$

2.4.4. Valor-F

El **Valor-F (*F-Score*)** se define como la media armónica de los valores de Precisión y Exhaustividad y por lo tanto, su valor se sitúa entre 0 y 1. Se trata de una métrica ampliamente utilizada en los ámbitos de recuperación de información y el procesamiento de lenguaje natural. Este valor nos permite medir y optimizar de manera balanceada la importancia de los valores de Precisión y Exhaustividad en nuestro sistema, logrando abarcar más información de manera sencilla.

$$F_1 = 2 \times \frac{Precision \times Exhaustividad}{Precision + Exhaustividad} \tag{2.4}$$

2.4.5. Métricas para múltiples candidatos

Como puede observarse, las métricas basadas en la matriz de confusión se encuentran limitadas a modelos binarios donde únicamente se mide el rendimiento respecto a una única categoría (positivo-negativo). Sin embargo, la subtarea de generación de sustitutos puede abordarse como una problemática con múltiples soluciones válidas, lo que obliga a modificar las métricas de evaluación. Un ejemplo de este planteamiento es la tarea compartida TSAR-2020 2.7.2, donde se permiten hasta 10 candidatos válidos. Para permitir una comparación más justa de los sistemas que proponen un número diferente de candidatos a la sustitución, los autores (Saggion et al., 2023) proponen unas métricas alternativas que se describen a continuación. Estas métricas no penalizan a los sistemas que devuelven menos candidatos del máximo, ya que todas las métricas de evaluación se aplican sobre un número fijo de los k-candidatos mejor clasificados. En el caso concreto de la TSAR-2020, se utilizaron diez métricas como métricas oficiales de la tarea compartida: ACC@1 (Exactitud), potential@k, accuracy@n@top1 y MAP@k, donde k $\in$ 3, 5, 10 y n $\in$ 1, 2, 3.

Potencial@k

El **Potencial**@k se define como el porcentaje de casos en los que al menos uno de los **k** sustitutos mejor clasificados también está presente en los datos de referencia.

Exactitud@k@top1

La **Exactitud**@k@top1 se define como el porcentaje de casos en los que al menos uno de los **k** sustitutos mejor clasificados coincide con el sinónimo sugerido con más frecuencia en los datos de referencia. Es importante señalar que Exactitud@1@top1 se denominó Exactitud@1 en (Stajner et al., 2022).

MAP@k

La métrica MAP se utiliza habitualmente para evaluar modelos de recuperación de información y sistemas de recomendación (Valcarce et al., 2020). En el contexto de la simplificación léxica, en lugar de utilizar una lista ordenada de documentos relevantes e irrelevantes, utilizamos una lista ordenada de sustitutos generados, que pueden coincidir (relevantes) o no coincidir (irrelevantes) con el conjunto de los sustitutos gold-standard. A diferencia de Precision@k, que sólo mide qué porcentaje de los k sustitutos mejor clasificados puede encontrarse entre los sustitutos gold-standard, MAP@k tiene en cuenta además la posición de los sustitutos relevantes entre los **k** primeros candidatos generados (es decir, si los candidatos relevantes están o no en las primeras posiciones).

2.5. Evolución de los sistemas de simplificación léxica

Como indican North et al. (North et al., 2023) en su reciente publicación, el ámbito de la simplificación léxica ha sufrido recientemente una profusión de novedades provocadas por los recientes avances en deep learning, en particular con la introducción de grandes modelos de lenguaje (en inglés *Large Language Models* o *LLMs*) y el aprendizaje a través de *prompts*. Estos modelos basados en LLMs han demostrado un rendimiento altísimo en los últimos años, así como un potencial de crecimiento relevante en varias subtareas.

Estos modelos basados en aprendizaje profundo han sustituido completamente a las aproximaciones anteriores de la simplificación léxica, basadas en modelos léxicos, modelos de reglas, modelos estadísticos o de word-embedding. Estos modelos previos funcionaban a través de transformaciones de las palabras candidatas a vectores que permitiesen calcular los mayores índices de similaridad respecto a la palabra original.

2.5.1. Modelos previos de simplificación léxica

En sus primeros años, la tarea de simplificación léxica automática se afrontaba mediante el uso de sinónimos que se decidían únicamente a partir de la frecuencia de palabras, asumiendo que a mayor uso de una palabra, más comprensible era esta(Devlin, 1998). Estos primeros modelos empezaron a combinarse con otras aproximaciones, en especial aquellas centradas en alteraciones sintácticas para tratar de simplificar la estructura de las frases sin perder información. Tampoco contaban con un proceso de identificación de palabras complejas (CWI), sino que consideraban la simplificación de todas las palabras. No sería hasta mucho más tarde, cuando comenzarían a utilizarse clasificadores para seleccionar qué palabras simplificar, evitando en gran medida la pérdida de información, por ejemplo mediante el uso de thresholds sobre las frecuencias de las palabras (Shardlow, 2013).

Estos modelos evolucionaron hasta otros que trataban de afinar más el proceso de CWI a partir del uso de lexicones específicos de complejidad. En su mayoría, estos lexicones se construían manualmente, identificando las palabras complejas a través de los distintos usuarios finales, como pueden ser niños o estudiantes. Estos modelos, así como otros basados en reglas o frecuencias estadísticas, fueron la norma hasta la llegada de las técnicas de aprendizaje automático. Estos funcionaban fundamentalmente en base a la idea de entrenar clasificadores binarios que discernían entre las palabras simples y complejas, aunque también se empleaban técnicas de regresión para entrenar modelos que medían el grado de complejidad de las palabras, de una forma más aproximada a los thresholds anteriores. Los algoritmos más frecuentes en estos modelos son los árboles de decisiones, los modelos supervisados basados en vectores (SVM) y algo después las redes neurales, en muchas ocasiones combinadas con recursos léxicos o morfológicos formando las diferentes etapas de un proceso mayor, conocido como "pipeline".

Aproximadamente a partir del año 2015, comienzan a utilizarse modelos

de incrustación de palabras (*word-embeddings* en inglés), que únicamente necesitan un corpus de texto sin anotar para ser entrenados, dejando atrás el uso de la mayoría de recursos que se usaban en las pipeline previas. En estos modelos de incrustación de palabras, cada palabra del vocabulario del corpus está representada por un vector único que contiene un número n de valores reales que describen la palabra en un espacio de características semánticas distribuido (Paetzold y Specia, 2017b). Estos modelos son capaces de detectar palabras sinónimas a otras seleccionando aquellas cuyo vector tenga una mayor similitud con el vector de la palabra compleja, utilizando una métrica llamada "similitud coseno". Estos modelos basados en word-embedding no solo mejoraban el rendimiento de los modelos anteriores, sino que su implementación era mucho más sencilla, al no depender de recursos externos.

2.5.2. Modelos de Lenguaje Enmascarados (MLMs)

A partir de los modelos de word-embedding, y como parte de su evolución, estos empezaron a combinarse con modelos generados por los LLM o por conjuntos de "puntuaciones de predicción"(*prediction scores* en inglés) generados previamente a través de los modelos LLM. De esta forma, utilizan modelos desarrollados con anterioridad, como Word2Vec, Sense2Vec o FastText en combinación con LLMs preentrenados. Sin embargo, estas aproximaciones no superaban en rendimiento a los enfoques más tradicionales preexistentes cuando estos se aplicaban directamente sobre léxicos para generar las palabras sustitutas (North et al., 2023).

Tras esto, en los últimos años el uso de LLMs pre-entrenados ha dado lugar a los Modelos de Lenguaje Enmascarados (MLMs de sus siglas en inglés) en el ámbito de la generación de sustitutos para la simplificación léxica. En el ámbito del español, (Qiang et al., 2020) desarrollan por primera vez un MLM para la tarea en español a través del lenguaje de modelo BERT (Bidirectional Encoder Representations from Transformers). Este modelo será llamado BERT-LS por los autores, y comenzará a usarse en la gran mayoría de los sistemas desarrollados para esta tarea, y en especial dentro del ámbito del TSAR-2022 (Saggion et al., 2015).

Este modelo de lenguaje enmascarado, el BERT-LS, se entrena a través de un proceso que enmascara aleatoriamente un porcentaje de los tokens de entrada, para a continuación predecir la palabra enmascarada a partir

de su contexto. De esta forma, si en el modelo se enmascara la palabra compleja en una frase, este MLM actúa como generador de candidatos de la palabra compleja para la simplificación léxica. Así, LS BERT-LS utiliza el MLM BERT original para la generación de candidatos de simplificación a través de enmascarar la palabra compleja de la frase original y utilizar esta versión para alimentar el modelo y obtener la distribución de probabilidad del vocabulario correspondiente a la palabra enmascarada. La ventaja de este método respecto a los utilizados con anterioridad en el estado del arte es que genera candidatos a la simplificación teniendo en cuenta toda la frase, no sólo la palabra compleja.

Por otro lado, el aprendizaje mediante *prompts* (en inglés *Prompt learning*) también ha surgido con fuerza en las aproximaciones más recientes de simplificación léxica. De hecho, los modelos basados en esta tecnología ocupan en este momento los mejores resultados para las tareas de sustitución de palabras en español, como en la tarea TSAR-2022. El aprendizaje mediante *prompts* consiste en introducir en un LLM una serie de entradas de texto con la descripción de la tarea específica y obtener los resultados devueltos por el modelo. Esto provoca que los *prompts* utilizados sean una de las claves en el rendimiento de un modelo y uno de los campos donde más se puede experimentar para desarrollar aproximaciones más precisas. Por ello, en la mayoría de artículos sobre estos modelos, los autores experimentan con varios *prompts*, variando su sintaxis y su léxico para seleccionar aquellas variantes con mejor rendimiento. Estos *prompts*, también llamados plantillas en ocasiones, porque se aplican sobre los diferentes objetos de un conjunto de datos, también se combinan con distintos parámetros y configuraciones en los casos en los que el LLM lo permite. De este modo, se puede realizar pruebas con subconjuntos de validación y configuraciones de ajuste fino, incluso en modelos con varios idiomas.

Varias propuestas de modelos de aprendizaje mediante *prompts* se describen en más detalle en la sección 3.8 del presente documento, donde destacan PromptLS (Vásquez-Rodríguez et al., 2022), generado a partir del dataset EASIER corpus, y la aproximación UniHD (Aumiller y Gertz, 2023) basada en GPT-3 de OpenAI.

Dentro del ámbito de la simplificación léxica, estos modelos recientes basados en aprendizaje profundo trabajan de forma asimétrica respecto a las diferentes subtareas existentes. Normalmente tienen una etapa de selección

de sustitutos (SS) mínima, que se realiza simultáneamente durante la propia generación de los sustitutos (SG). Esto se debe fundamentalmente a que las predicciones de un LLM preentrenado ya incluyen un filtro de palabras que normalmente serían descartadas en pasos posteriores. De la misma forma, las técnicas de puntuación para ordenar (SR, por sus siglas en inglés) de palabras candidatas no suelen implementarse, utilizando las referencias del propio LLM.

Por todo esto, podemos concluir que los enfoques de aprendizaje profundo han proporcionado nuevos avances en el campo de la simplificación léxica, al igual que ha ocurrido en otros muchos ámbitos del PLN. Como se podrá ver en detalle más adelante, los MLM son ahora la aproximación utilizada en la mayoría de publicaciones recientes sobre simplificación léxica, superando el rendimiento de todos los demás enfoques en las métricas utilizadas. Estas aproximaciones MLM, u otros modelos basados en LLM, han sustituido las técnicas anteriores, además de alterar el peso entre las diferentes subtareas que hasta ahora se venían utilizando, sustituyendo a varias de ellas por un único paso. Es lógico pensar que los modelos de aprendizaje mediante *prompts* y otros MLM serán aún más utilizados en el futuro inmediato y que aún cuentan con un amplio margen de mejora, dado el bajo número de aproximaciones presentadas por el momento.

2.5.3. Aprendizaje basado en el uso de instrucciones (*prompts*) en LLMs

En comparación con los modelos previos existentes, el uso de instrucciones (*prompting* en inglés) es mucho más sencillo, ya que no introduce grandes cantidades de parámetros adicionales ni requiere la inspección directa de las representaciones de un modelo. Sin embargo, aunque esta técnica se ha utilizado para obtener resultados relevantes en multitud de ámbitos, normalmente se basan en instrucciones creadas manualmente a partir de la intuición del experimentador. Estas instrucciones manuales pueden no ser óptimas debido a que los LLM pueden haber aprendido el conocimiento objetivo de contextos sustancialmente diferentes durante su creación respecto al momento actual (Jiang et al., 2020).

La obtención de resultados a partir de los LLM es bastante diferente a otras bases de conocimiento utilizadas en el ámbito del PLN. Mientras que los modelos clásicos se basan en consultas estructuradas y definidas por un

esquema de proceso, los LLM generan su información a través de consultas formadas por instrucciones en lenguaje natural.

El uso de instrucciones tiene también varias ventajas, especialmente por su escasa necesidad de parámetros adicionales, lo que rebaja sensiblemente los requisitos técnicos y computacionales para realizar tareas extensas. También implica una línea base potente, que suele obtener resultados competitivos en planteamientos sencillos, puesto que es capaz de extraer mucha información de una instrucción de texto.

Min, et al. (Min et al., 2020) clasifican las diferentes aproximaciones basadas en instrucciones en tres categorías diferentes, que se resumen a continuación.

Instruction based learning: También llamado «aprendizaje a partir de instrucciones y demostraciones», estos modelos utilizan tareas y ejemplos para guiar al LLM en la realización de la tarea. Por ejemplo, (Schick y Schütze, 2021) utilizan un LLM generativo (GPT2-XL) para generar frases a partir de un token que enmascara la palabra clave. Es el más utilizado en modelos de gran tamaño, como GPT-3.

Template based learning: El aprendizaje basado en plantillas utiliza ejemplos etiquetados que se transforman en texto natural a través de plantillas. Estas plantillas suelen incluir espacios vacíos que se rellenan con información sobre los ejemplos o resultados de los modelos. Se utilizan frecuentemente en LLMs enmascarados o de tamaño relativamente pequeño que no están ajustados a la tarea objetivo. La aproximación de UniHD (Aumiller y Gertz, 2023) para la TSAR 2022 es un ejemplo de uso de plantillas.

Proxy-task based learning: El aprendizaje basado en tareas-proxy consiste en el uso de instrucciones que transforman los ejemplos de la tarea objetivo en ejemplos de tareas-proxy. En este caso, los LLM se adaptan a las tareas-proxy antes de aplicarse a la tarea objetivo. Estas actúan como entradas de información al modelo y se componen de una instrucción junto con la entrada de la tarea original. La principal diferencia entre los métodos anteriores y este método es el uso de tareas supervisadas en lugar de un modelado del lenguaje. Por ejemplo, Li et al. (Li et al., 2020) utilizan esta aproximación para afrontar el reconocimiento de entidades nombradas (NER, por sus siglas en inglés) utilizando clasificadores binarios independientes que identifican los tokens iniciales y finales de las instrucciones generadas.

2.6. Recursos existentes para simplificación léxica en español

A continuación se describen los principales conjuntos de datos desarrollados durante los últimos años y disponibles en el estado del arte para la tarea de simplificación léxica en español. Se incluyen conjuntos de datos que forman parte de tareas compartidas, que incluyen soluciones destinadas a evaluar sistemas, pero también modelos de datos pre-entrenados que pueden utilizarse de forma libre.

2.6.1. Corpus EASIER

El corpus EASIER (Alarcon, Moreno, y Martínez, 2023) es un recurso creado para utilizarse en métodos de simplificación léxica para procesar textos en español de ámbito general. Contiene 260 documentos anotados que incluyen 8.155 palabras etiquetadas como complejas y 5.130 palabras con al menos una propuesta de sinónimo contextual. Su objetivo es mejorar las tareas de identificación de palabras complejas (CWI) y generación de sustituos (SG/SS), en especial en aquellos modelos destinados a un público con discapacidad intelectual. Para su generación se ha utilizado un proceso de anotación y evaluación mediante expertos lingüistas especializados en directrices de fácil lectura y lenguaje sencillo, para discernir entre palabras complejas y simples. Además, los autores han realizado diferentes experimentos para validar EASIER a través de varias técnicas, incluyendo un estudio experimental con 45 participantes, contando con personas con discapacidades cognitivas para ello. Esta evaluación se realizó mediante las métricas más utilizadas en el ámbito del aprendizaje automático, como la exactitud, la precisión y el valor-F.

Este recurso está integrado en la plataforma EASIER[2], una herramienta más completa que ayuda a las personas con deficiencias cognitivas y discapacidad intelectual a leer y comprender textos con mayor facilidad. Los autores afirman que en futuros trabajos desarrollarán también una ampliación de este recurso para dominios específicos, como textos jurídicos o sanitarios, entre otros.

También se proporciona una versión reducida del corpus, llamada EASIER-500 (Alarcon, Moreno, y Martínez, 2021), que comprende 500 instancias,

[2]https://github.com/ralarcong/EASIER_AnnotationTool

cada una con una frase, una palabra compleja y tres sustituciones sustituciones contextuales sugeridas por un lingüista experto. Actualmente EASIER se encuentra disponible de forma abierta bajo licencia Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License CC-BY-NC-SD-4.0.[3]

2.6.2. ALEXSIS: A Dataset for Lexical Simplification in Spanish

ALEXSIS es un conjunto de datos para la evaluación comparativa de la simplificación léxica en español, presentado por Ferrés y Saggion(Ferrés y Saggion, 2022) en 2022. Este contiene 381 instancias, cada una compuesta de una frase/contexto, una palabra compleja objetivo y 25 sustitutos candidatos. Para construirlo, los autores extrajeron las frases y palabras complejas del conjunto de datos CWI Shared Task 2018 para español, que contiene contiene 7.015 palabras complejas en las que 4.712 son palabras únicas (Yimam et al., 2018). Para ello extrajeron un subconjunto de 588 pares ≪sentencia, palabra compleja≫ de los archivos Train, Dev y Test del conjunto de datos CWI.

Para extraer las instancias los autores utilizaron un criterio basado en fue que la palabra compleja objetivo fuese anotada como tal por 5 o más anotadores nativos y que no fuera una expresión formada por varias palabras o que incluyese al menos una letra mayúscula. Después, a partir del conjunto de 588 pares, llevaron a cabo un proceso de evaluación manual mediante 2 lingüistas con el objetivo de decidir si la palabra compleja objetivo era ≪simplificable≫ por los expertos De este proceso se obtuvieron 3 conjuntos según distintos criterios; un conjunto de 256 pares en los que ambos revisores estaban de acuerdo en que la palabra compleja es simplificable, un conjunto de 113 pares en los que ambos revisores estaban de acuerdo en que la palabra compleja no era simplificable, y un conjunto de 219 pares en los que existía desacuerdo entre los revisores sobre la simplificación. Este proceso de revisión se realizó con la ayuda de diccionarios y tesauros en línea. Respecto a las palabras complejas objetivo, existen en el conjunto de datos 356 palabras objetivo diferentes, de las cuales 333 palabras aparecen una única vez, 21 palabras aparecen dos veces, y 2 palabras aparecen tres veces. Hay un total de 9.524 sustituciones en el conjunto de datos, para un total

[3] https://github.com/lurmoreno/easier_corpus

de 3.918 sustituciones diferentes.

A diferencia de otros conjuntos de datos anteriores para la evaluación de la simplificación léxica en español, ALEXSIS que incluye información adicional para realizar rankings de simplicidad Léxica (*Lexical Simplicity Ranking*) e incluye un mayor número de resultados válidos para cada palabra objetivo. Actualmente ALEXSIS se encuentra disponible de forma abierta bajo licencia Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License CC-BY-NC-SA-4.0.[4]

Respecto a otros recursos, ALEXSIS es el primer conjunto de datos para la evaluación de la simplificación léxica en español que incluye información potencialmente útil para realizar rankings de simplificación. Además, cuenta con un mayor número de sinónimos por instancia en comparación con EASIER/EASIER-500 (Ferrés y Saggion, 2022).

2.6.3. ALexS 2020 / VYTEDU-CW corpus

Creado para la primera edición del workshop ALexS (*Task on Lexical Analysis at SEPLN*) (Ortiz-Zambranoa y Montejo-Ráezb, 2020), una tarea compartida de análisis léxico que forma parte del Congreso Intencional de la Sociedad Española de Procesamiento del Lenguaje Natural (SEPLN). Está enfocado en la tarea de identificación de palabras complejas (CWI) y su conjunto de datos se ha generado a partir del conjunto original VYTEDU-CW (Ortiz Zambrano et al., 2019). Desde este conjunto original, formado por vídeos y transcripciones del ámbito educacional, los autores han generado un conjunto de datos anotados formado por 723 palabras complejas, contextualizadas en diferentes documentos e identificadas y etiquetadas por 430 estudiantes.

Únicamente tres equipos presentaron sus modelos en el ALexS 2020, lo que sumado a sus mediocres resultados obtenidos (North, Zampieri, y Shardlow, 2023) le resta relevancia en el estado del arte respecto a otros conjuntos de datos, como ALEXSIS o EASIER.

2.6.4. BERTIN / RoBERTa

BERTIN(De la Rosa et al., 2022) es un conjunto de modelos de aprendizaje en profundidad para el español a partir de los modelos BERT (*Bidi-*

[4]https://github.com/LaSTUS-TALN-UPF/ALEXSIS

rectional Encoder Representations from Transformers)(Devlin et al., 2018), basados en redes neuronales para el pre-entrenamiento con representación de contexto y aprendizaje de secuencia semi-supervisado. desarrollada por Google. BERT está pre-entrenado con 800 millones de palabras del BooksCorpus y 2.500 millones de palabras de Wikipedia en inglés, una cantidad enorme de datos en comparación con otros modelos. Con ellos, se entrena un modelo base que aprenda a interpretar el lenguaje en general, en lugar de implementar distintos modelos específicos que afronten cada tarea por separado. Una vez hecho esto, se añade a BERT varias capas adicionales para especializarlo en una tarea en particular mediante un proceso denominado *fine-tuning*.

En concreto, BERTIN trabaja con una configuración de BERT conocida como RoBERTa (Liu et al., 2019), que utiliza el enmascaramiento dinámico y un BPE a nivel de byte (*Byte-level Byte-Pair-Encoding*) como tokenizador, así como otras técnicas de preentrenamiento, para diferenciarse del resto. Como conjunto de datos, BERTIN utiliza mC4, una variante multilingüe del corpus C4 (*Colossal Clean Crawled Corpus*)(Xue et al., 2020) cuya sección en español contiene alrededor de 416 millones de documentos y 235.000 millones de palabras.

Los autores de BERTIN han realizado varios experimentos con diferentes métodos de muestreo sobre la versión en español de mC4 (mC4-es), incluyendo una técnica de muestreo de perplejidad que permite el preentrenamiento de modelos de lenguaje de forma mucho más eficiente. Su código fuente de está disponible bajo licencia Apache 2.0, incluyendo varias versiones mejoradas para tareas específicas.

2.6.5. Modelos GPT

Además de BERT, otros modelos de lenguaje pre-entrenados han mostrado un gran potencial para el procesamiento del lenguaje natural (Vaswani et al., 2017). Entre estos, destacan los modelos GPT, modelos que han aumentando su número de parámetros hasta alcanzar los cientos de miles de millones en sus últimas versiones, GPT-3 y GPT-4 (Koubaa, 2023). Estos cuentan a su vez con sistemas conversacionales de dominio abierto como DialoGPT, InstructGPT y ChatGPT, creados a partir de versiones afinadas de sus modelos para tareas específicas. Desde su publicación, los modelos de lenguaje GPT (Brown et al., 2020) de OpenAI han permitido el uso ex-

tendido de LLMs de propósito general en multitud de tareas, en especial en aquellas que contaban con recursos más escasos o de difícil acceso. Pese a tratarse de modelos optimizados para su uso en inglés, muchos de ellos son lo suficientemente robustos como para generar buenos resultados en una gran variedad de idiomas, incluyendo el español, incluso utilizando textos o instrucciones en inglés(Aumiller y Gertz, 2023). En el ámbito específico de la simplificación de textos, desde su reciente aparición los enfoques basados en GPT-3 superan el rendimiento de todos los demás enfoques cuando se somete al aprendizaje basado en instrucciones (North et al., 2023).

2.7. Foros de evaluación

En esta sección se analizan los principales foros de evaluación sobre la tarea de simplificación léxica en español. Este análisis resulta de especial interés debido a que gran parte de los avances en investigación, así como las publicaciones científicas sobre simplificación léxica se enmarcan en el contexto de estos foros.

2.7.1. Complex Word Identification (CWI) Shared Task 2018

La tarea compartida de identificación de palabras complejas (CWI)(Yimam et al., 2018), celebrada durante NAACL-HLT'2018, está formada por conjuntos de datos multilingües y multigénero divididos en cuatro secciones; inglés, alemán, español y multilingüe. Abarca la tarea de dos formas distintas, mediante clasificación binaria y clasificación probabilística. En el caso de la clasificación binaria, los modelos debían clasificaran las palabras como complejas (1) o simples (0), mientras que en la clasificación probabilística debían asignar un valor de probabilidad de que las palabras del contexto fueran complejas.

Los conjuntos de datos en inglés abarcan tres géneros textuales; noticias escritas por profesionales, noticias escritas por aficionados y artículos de Wikipedia. La tarea de anotación en español contiene únicamente artículos de Wikipedia y, a diferencia de las tareas de anotación del resto de idiomas, sus anotaciones proceden casi exclusivamente de anotadores nativos. Cuenta con 13.750 instancias para entrenamiento, así como 2,233 de pruebas. Un total de 12 equipos presentaron sus resultados en distintas combinaciones de tareas, incluyendo 8 de ellos para la tarea en español.

2.7.2. TSAR-2022 Shared Task on Lexical Simplification

Esta tarea compartida de simplificación léxica presenta tres conjuntos de datos similares en tres idiomas diferentes: inglés, portugués de Brasil y español. Se realizó en 2022 como parte de la Conference on Empirical Methods in Natural Language Processing (EMNLP) por primera vez, y tanto sus resultados como los conjuntos de datos son de acceso público [5]. Esta tarea se explica en detalle en el capítulo 3.

El planteamiento de la tarea es sencillo, aunque su funcionamiento es algo diferente a otras tareas que abordan el mismo tema. En este caso, dada una frase que contiene una palabra compleja, los sistemas participantes deben devolver una lista ordenada de sustitutos válidos de la palabra compleja que hagan más sencilla la comprensión de la frase en su contexto original. La lista de palabras más sencillas (hasta un máximo de 10) enviada por el sistema debe estar ordenada según la confianza del sistema en su predicción, con las mejores predicciones primero. Hay que tener en cuenta que el conjunto de datos de la tarea ya indica cuales son las palabras complejas a sustituir, por lo que esta tarea no afronta la primera subtarea Complex Word Identification (CWI). Para el conjunto de datos en español los organizadores de la tarea han utilizado ALEXSIS, desarrollado por Daniel Ferres y Horacio Saggion(Ferrés y Saggion, 2022).

De entre todos los modelos presentados, hay que destacar la de Aumiller y Gertz (Aumiller y Gertz, 2023), pues obtiene los mejores resultados tanto en inglés como en español, mediante una aproximación a través del modelo GPT-3.

A fecha de conclusión de este trabajo, se ha anunciado una nueva edición de la tarea TSAR como parte RANLP-2023, aunque aún no se han publicado sus trabajos. El conjunto de datos y los métodos de evaluación no han sufrido cambios.

2.7.3. ALexS Workshop on Lexical Analysis at SEPLN.

(Ortiz-Zambranoa y Montejo-Ráezb, 2020) En 2019 los foros de evaluaciones TASS e IberEval se unen para formar un único foro llamado IberLef. Como parte de su segunda edición, se inicia una tarea compartida centrada en la sub-tarea de simplificación léxica de identificación de palabras com-

[5] https://github.com/LaSTUS-TALN-UPF/TSAR-2022-Shared-Task

plejas (CWI). Más de siete equipos se unieron a la tarea pero sólo tres de ellos presentaron finalmente la descripción de sus sistemas; UDLAP (Rico-Sulayes, 2020), Vicomtech(Zotova et al., 2020) y HULAT(Alarcón, Moreno, y Martínez, 2020).

Para la tarea, los autores crearon un corpus ad hoc a partir del corpus original VYTEDU-CW, el cual ya se ha descrito en la sección 2.6.3, dentro de los recursos existentes. Los pocos resultados obtenidos en la tarea arrojan resultados pobres en cuanto a precisión, debido a las diferencias entre el corpus de entrenamiento y el de pruebas, que pertenecen a dominios distintos. Los modelos presentados utilizan algoritmos supervisados (SVM) o basados en la frecuencia de aparición de las palabras de forma no supervisada.

2.7.4. SimpleText: Automatic Simplification of Scientific Texts

La tarea compartida SimpleText forma parte de la iniciativa CLEF, que promueve la evaluación sistemática de los sistemas de acceso a la información. En su caso concreto, SimpleText aborda la tarea de simplificación de textos en el contexto de la información científica. Para ello utiliza un conjunto de datos formado por resúmenes de literatura científica y solicitudes de divulgación científica con el objetivo de crear un resumen simplificado de múltiples documentos científicos basado en una consulta de divulgación científica que proporcione al usuario una visión general accesible de este tema específico (Ermakova et al., 2022). Su última edición ha tenido lugar en 2023 y ha contado con tres tareas distintas; seleccionar pasajes para incluir en el resumen, identificar y explicar conceptos difíciles y por último reescribir el texto científico. Cada una cuenta con su propio conjunto de datos, generado a partir de bases de datos de textos científicos, sobre los que se han seleccionado manualmente los temas. Los resultados de esta última edición aún no se han hecho públicos a fecha de redacción de este trabajo.

2.7.5. FinToc-2022: Financial Document Structure Extraction

La tarea compartida FinTOC-2022(Kang et al., 2022a) se organizó como parte del 4th Financial Narrative Processing Workshop (FNP 2022), celebrado conjuntamente con la 13th Edition of the Language Resources and Evaluation Conference (LREC 2022). Aunque de forma tangencial, el objetivo de esta tarea también forma parte de la simplificación léxica. Está

centrada en la extracción de tablas de contenido (TOC) en documentos de inversión mediante la detección de los títulos de los documentos y su organización jerárquica. Cuenta con tres subtareas, cada una en un idioma, incluyendo una para documentos en español. Para generar los conjuntos de datos que se utilizan en la tarea, los autores han seleccionado los documentos específicamente para que incluyan gran variedad de estructuras, a partir del corpus Fin-T-esp(Moreno-Sandoval, Gisbert, y Montoro, 2020) en el caso de los documentos en español. Este conjunto de datos abarca 90 ficheros pdf, con una media de 158 páginas por documento, anotados por tres anotadores de forma manual. De los modelos planteados, para los datos en español destaca el uso de una red neural pre-entrenada con el conjunto de datos PubLayNet, que presenta resultados ligeramente superiores al resto, basados en árboles de decisiones (Cassotti et al., 2022).

2.8. Conclusiones

Desde su reciente llegada, los modelos de aprendizaje profundo han destacado por su eficaz rendimiento en las diferentes sub-tareas que abarcan la simplificación léxica. En las diferentes tareas compartidas que se han realizado en los últimos años sobre simplificación, los modelos de lenguaje enmascarado tienen un papel destacado y muestran los mejores resultados en todas las métricas estándar. Una visión general hace obvio que entre las mejores aproximaciones recientes destacan los lenguajes de modelo basados en BERT así como los modelos GPT, sobretodo GPT-3. Esto ocurre no solo en las tareas que utilizan textos en inglés, los más extendidos del estado del arte, sino también en el dominio específico de los textos en español. Esto revela una facilidad de estos modelos para adaptarse a otros idiomas y abre una ventana de oportunidad para estrechar la diferencia que ha existido en los últimos años en los rendimientos sobre los distintos idiomas.

También es importante tener en cuenta que estos modelos más recientes afrontan la tarea de simplificación léxica de una forma algo distinta, desdibujando los límites que existían previamente entre las diferentes sub-tareas que la formaban. Algunas sub-tareas, como la selección de sustitutos (SS), la ordenación de candidatos (SR) o incluso la adaptación al contexto se realizan de forma interna a los modelos, simplificando enormemente el proceso, pero dificultando enormemente la posibilidad de evaluar sus rendimientos

por separado.

Estos hechos evidencian la necesidad de experimentar con los modelos de lenguaje enmascarado para mejorar su rendimiento y resolver las problemáticas de un área de investigación recién abierta, donde apenas se han extraído las primeras conclusiones. Es necesario avanzar en conocimiento respecto a tareas intrínsecas de los LLM, cómo la creación de instrucciones (*prompting*), el ajuste (*fine-tuning*) o la personalización a grupos de usuarios particulares.

Para tratar esto en el ámbito de la simplificación léxica en español, y específicamente en la generación de sustitutos (SS y SR), el presente trabajo propone abordarlo a través de la tarea compartida TSAR 2.7.2, al considerarla la más completa y la que cuenta actualmente con los modelos más modernos con los que poder compararse. Por ello, en primer lugar, se realizará un análisis en profundidad de los modelos presentados con mejores resultados, para identificar aquellas técnicas más adecuadas y con mejor rendimiento. Una vez establecido un marco de referencia, se experimentará con los diferentes parámetros disponibles, en especial con aquellos que permitan una mayor especificidad al dominio en español, con la intención de descubrir que factores influyen en su rendimiento y de qué forma, así como fortalezas y debilidades de los mismos.

Capítulo 3

Generación de sustitutos en la tarea compartida TSAR-2022

3.1. Introducción

En este capítulo se presentan los elementos de la tarea compartida TSAR (*Text Simplification, Accessibility, and Readability Workshop*) que formarán el marco de trabajo sobre el que experimentar el problema de la simplificación léxica en español. Inicialmente se describe el funcionamiento de la tarea, así como el conjunto de datos creado por sus autores y utilizado para la evaluación de los modelos. A continuación, se presentan las diferentes métricas utilizadas para medir el rendimiento de los modelos sobre el conjunto de datos. Finalmente, se realiza un exhaustivo análisis de las aproximaciones presentadas por los participantes en la última edición, desarrollando una serie de conclusiones sobre las técnica y herramientas más utilizadas y aquellas que presentan un mayor rendimiento, como son las basadas en los modelos del lenguaje.

3.2. Propósito de la tarea TSAR

Esta tarea compartida aborda la simplificación léxica multilingüe mediante el desarrollo de métodos innovadores para avanzar en el estado del arte de la simplificación léxica para inglés, portugués (brasileño) y español.

Los trabajos presentados en su última edición exploran diversas arquitecturas e indican nuevos puntos de referencia dentro de la simplificación léxica y, pese al predominio de los sistemas en inglés, los resultados para el español son también muy relevantes y confirman una tendencia clara a favor de los LLMs.

La tarea plantea sistemas que, dada una lista de palabras complejas en sus frases, sean capaces de simplificar dichas palabras en su contexto específico. En referencia a las sub-tareas descritas en la sección 2.3.1, los sistemas deben realizar los pasos 2-5 para generar, seleccionar, clasificar y adaptar al contexto sustitutos de una palabra compleja dada en una frase. Esto conlleva investigar estrategias de mejora del contexto en combinación con métodos de clasificación y selección de sustituciones.

Descripción de la tarea

Con la excepción de algunas diferencias a la hora de generar el conjunto de datos, la tarea funciona de la misma forma para los tres idiomas, así que aunque este trabajo esté centrado en el español, lo explicado aquí es aplicable al resto de idiomas de la tarea.

Dada una una frase/contexto y una palabra compleja en ella, el sistema debe proporcionar sustitutos de la palabra compleja objetivo que facilite la comprensión de la frase. Para ello, los sistemas pueden ofrecer hasta 10 sustitutos diferentes, aunque deben ser ordenados de la más a la menos adecuada/simple, sin aceptar empates entre palabras. A diferencia de otras tareas similares, no se proporciona a los participantes un conjunto de entrenamiento, pero sí varios ejemplos de prueba en cada lengua, pues se espera que los participantes hagan uso de recursos externos para crear sus sistemas, en lugar de entrenar modelos.

Durante el proceso de creación de los conjuntos de datos en inglés y español, las frases/contextos y las palabras (complejas) objetivo se seleccionaron a partir de conjuntos de datos anteriores, concretamente los presentados en la tarea compartida BEA-2018 (Yimam et al., 2018) sobre identificación de palabras complejas (CWI). En dichos conjuntos de datos, las palabras complejas se marcaron mediante anotadores humanos vía crowdsourcing, con 10 hablantes nativos y 10 no nativos, tanto en inglés como en español. Estos anotadores marcaban las palabras como difíciles de entender en un contexto determinado, específicamente un párrafo que contiene varias fra-

ses, quedando anotadas como complejas en los conjuntos de datos finales aquellas que eran marcadas al menos una vez. Debido a que las frases tenían a menudo varias palabras marcadas como complejas, los organizadores escogieron solo una de las palabras complejas en cada frase seleccionada para compilar los conjuntos de datos de la tarea compartida TSAR-2022. De esta forma, se facilita la tarea a los participantes, ya que deben tener en cuenta cómo encaja en el contexto el sustituto más simple propuesto. Es decir, si conserva o no el significado original, en lugar de tener en cuenta las interacciones entre los sustitutos propuestos de diferentes palabras objetivo dentro de la misma frase. A continuación se muestra un ejemplo de frase incluida en el conjunto de datos, tal y cómo se explica en (Saggion et al., 2023).

Frase/Contexto: Floreció en la época clásica y tenía una ***reputada*** escuela de filosofía.
Palabra objetivo: reputada
Sustitutos ordenados (gold): prestigiosa:6, famosa:4, reconocida:2, afamada:2, conocida:2, renombrada:2, respetada:2, prestigioso:1, muy reconocida:1, valorada:1, acreditada:1, prestigiada:1

Ante cada una de las frases, el sistema debe generar un conjunto de nombres de sinónimos candidatos propuestos, donde no se permiten palabras complejas. Además, estos candidatos deben tener la misma inflexión morfológica que la palabra compleja en la frase original, o se considerarán como erróneos. A partir de estas respuestas, los modelos se evalúan a través de un *Evaluation Benchmark* ofrecido por la organización de la tarea y que mide diez métricas distintas, descritas en detalle en 2.4.

3.3. Análisis del conjunto de datos ALEXSIS

Los conjuntos de datos utilizados por las aproximaciones del TSAR-2022-Shared-Task, junto al script de evaluación, están disponibles en un repositorio de github[1] bajo licencia CC-BY-NC-SA-4.0.

El objetivo de los autores con la creación del conjunto de datos es superar el problema de la tarea de simplificación léxica debido a la ausencia de conjuntos de datos fiables para la evaluación automática en el estado del arte actual, especialmente en idiomas distintos al inglés. Este problema tiene su origen en el elevado coste en tiempo y recursos para la evaluación mediante

[1] https://github.com/LaSTUS-TALN-UPF/TSAR-2022-Shared-Task

Idioma	Sustitutos por palabra objetivo			Test	Prueba	Total
	Min	Max	Avg			
EN	2	22	10.55	373	10	383
ES	2	19	10.28	368	12	380
PT-BR	1	16	8.10	374	10	384

Tabla 3.1: Casos anotados del conjunto de datos ALEXSIS

expertos humanos, que anteriormente se ha mitigado mediante el uso de métodos automatizados, cuyo coste es inferior, pero también su rendimiento.

De la misma forma, los autores presentan un conjunto de datos multilingüe en inglés, español y portugués de Brasil, con el objetivo de facilitar la evaluación de sistemas de simplificación léxica en corpus multilingües. De esta forma, dado que todos ellos son formados y anotados de forma comparable siguiendo las mismas prácticas, se facilitan los análisis comparativos entre lenguas, dentro de la tarea. Incluyendo el rendimiento de sistemas desarrollados para aplicarse sobre conjuntos en distintos idiomas.

Para obtener una lista de palabras sustitutas que simplifiquen la frase para cada palabra objetivo, se utilizaron trabajadores en crowdsourcing para proponer sustitutos a una serie de frases seleccionadas (386 en inglés y 381 en español). Mientras que en inglés esta tarea de anotación se realizó mediante Amazon Mechanical Turk[2], en español utilizó la plataforma Prolific[3]. Las directrices utilizadas para la anotación en español se tradujeron después al inglés y al portugués con una edición mínima para garantizar que la tarea siguiera siendo la misma en todos los idiomas. El número total de casos anotados, el número mínimo, el máximo y la media de sustitutos más sencillos propuestos por palabra meta en cada lengua se indican en la tabla 3.1. Las frases anotadas en cada idioma se dividieron en conjuntos de datos de evaluación y de prueba y están disponibles bajo la licencia Creative Commons AttributionNonCommercial-ShareAlike 4.0 International License (CC-BY-NC-SA-4.0).

Dada la metodología utilizada para la creación de candidatos, es necesario utilizar unas métricas que permitan una comparación equilibrada de los sistemas que proponen un número diferente de candidatos a la sustitución y de esta forma no penalizar a los sistemas que devuelven menos candidatos. Para ello, todas las métricas de evaluación se aplican sobre una cantidad

[2]https://www.mturk.com/
[3]https://www.prolific.co/

Frase	*Sufrio una importante reducción en su capacidad para poder* ***acogerse*** *a las normas de la FIFA para los estadios de futbol.*
Anotación	adaptarse (6), refugiarse (2), apegarse (2), ampararse (2), aceptar (2), incorporarse (2), sumarse, recurrir, obedecer, cumplir con, asimilarse, aplicarse, amparar, admitirse, aceptarse

Tabla 3.2: Ejemplo de frase y anotaciones de ALEXSIS en español. Palabra clave indicada en negrita.

fija de los k-candidatos mejor clasificados. De la misma forma, para tener en cuenta diversos aspectos del rendimiento de los sistemas, se utilizan diez métricas como métricas oficiales de la tarea compartida: ACC@1, MAP@k, potential@k, accuracy@n@top1 donde $k \in \{3, 5, 10\}$ y $n \in \{1, 2, 3\}$. Hay que tener en cuenta que ACC@1, MAP@1 y Potential@1 dan los mismos resultados por definición y por ello se utiliza únicamente ACC@1, en lugar de los tres. Estas métricas de describen en detalle en la sección 3.4.

En esta sección se resume la información del artículo **ALEXSIS: A Dataset for Lexical Simplification in Spanish** (Ferrés y Saggion, 2022), en la que se describe la compilación del conjunto de datos para el español, así como el artículo **Lexical simplification benchmarks for English, Portuguese, and Spanish** (Stajner et al., 2022), que presenta el proceso general para la tarea. Tras ello, se realiza un análisis en profundidad de sus diferentes características, destacando los datos más relevantes, y finalmente se presentan algunas conclusiones al respecto.

El conjunto de datos ALEXSIS de simplificación léxica en español contiene 381 instancias, cada una compuesta por una frase, una palabra compleja (objetivo) y 25 sustituciones candidatas. A diferencia de otros conjuntos de datos de tareas similares, en ALEXSIS las frases no están tokenizadas, sino que se presentan en texto plano. Tanto las frases como las palabras complejas de este conjunto de datos han sido extraídas del conjunto de datos CWI Shared Task 2018 para español, que contiene 7.015 palabras complejas de las cuales 4.712 son palabras simples. A partir de ahí, se extrajeron 588 pares (sentencia, palabra compleja) de los archivos Train, Dev y Test del conjunto de datos CWI para español, según un criterio específico. La tabla 3.2 muestra una frase del conjunto a modo de ejemplo.

Las palabras complejas de cada frase fueron anotadas como palabra compleja por 5 o más anotadores nativos y, además, se excluyeron aquellas consistentes en una expresión de varias palabras o una palabra que tuviera al menos una letra mayúscula. Después, los autores llevaron a cabo un proceso

de evaluación manual, en el que participaron 2 lingüistas, para decidir si cada palabra compleja era simplificable en su contexto o no. Este proceso de revisión fue realizado con la ayuda de diccionarios y tesauros.

Una vez seleccionadas las palabras complejas, para obtener los sinónimos para cada palabra compleja, los autores utilizaron la plataforma de crowdsourcing *prolific.co* para contratar anotadores con unos requisitos mínimos de competencia lingüística y educación (hablar español con fluidez y tener un título universitario (BA/BSc/otro) o de posgrado (MA/MSc/MPhil/otro)). Estos anotadores debían proponer una sola palabra que fuera el sinónimo válido más sencillo o que sustituyera a la palabra compleja en su contexto e hiciera más fácil de entender. la frase. Además, en caso de no poder utilizar una sola palabra, se les permitió el uso de varias palabras.

Para cada frase, un conjunto de 25 anotadores propuso un sustituto más sencillo, divididos en 3 grupos de 128 frases que se enviaron a 75 anotadores diferentes. Los datos demográficos de los 75 anotadores son los siguientes:

- **Sexo:** Mujeres (47), Hombres (28);
- **Rangos de edad (años):** 20-30 (54) 31-40 (17) 41-50 (2) 50-59 (1), Desconocido (1);
- **Nacionalidad:** Argentina (1), Grecia (1), Italia (2), Venezuela (2), Portugal (4), España (6), Chile (13), México (45), Desconocido (1).

Una vez finalizado el proceso de anotación, los autores decidieron eliminar 2 casos donde la palabra compleja estaba repetida en la frase, y una frase que tenía un error tipográfico, dejando el número final de frases del conjunto de datos en 381 instancias. En ellas hay 356 palabras objetivo diferentes, de las cuales 333 palabras aparecen una única vez, 21 palabras aparecen dos veces, y 2 palabras se repiten tres veces.

Así mismo, hay en el conjunto de datos un total de 9.524 sustituciones y, tras unir las sustituciones repetidas en cada frase, el total es de 3.918 sustituciones diferentes. Tras una revisión final, las sustituciones correctas quedaron en 9.064, siendo descartadas el resto por sustituciones incorrectas (137), sustituciones dudosas (93) o sustituciones iguales a la palabra compleja (230).

Por otra parte, ALEXSIS tiene algunos problemas que no han sido resueltos y que deben tenerse en cuenta a la hora de evaluar herramientas con él. Como bien indican los propios autores, los conjuntos de datos en inglés

y español proceden de la misma fuente y cubren un solo género de texto. Además, las sustituciones propuestas en ALEXIS representan los sinónimos más sencillos, según los anotadores de la plataforma de crowdsourcing, en lugar de anotadores expertos en la materia. El elevado número de anotadores por caso (25) mitiga este problema en cierta medida, ya que ofrece la posibilidad de clasificarlos en función del número de veces que fueron sugeridos. En el caso del conjunto de datos en español, este problema puede verse reforzado por el desequilibrio existente entre las nacionalidades de los supervisores, donde los supervisores de México superan en número a la suma de todos los demás, pudiendo provocar una desproporción a partir de las variedades lingüísticas y sociolectos existentes en los diversos territorios. Una línea de trabajo futura para ALEXSIS sería tratar de comprobar hasta que punto estos desequilibrios influyen en el rendimiento o la especificidad de los modelos evaluados con él, por ejemplo, replicando el proceso con documentos de otros géneros de texto, o usando anotadores expertos.

3.4. Métricas utilizadas en TSAR-2022

Pese a que las métricas de TSAR-2022 ya se han visto como parte del estado del arte en el capítulo previo (Sección 2.4), aquí se revisan de nuevo para poder contextualizarlas en la tarea y en los resultados obtenidos por sus participantes.

Como hemos visto al describir las métricas de uso general en la tarea, el uso de múltiples candidatos en la tarea requiere de métricas que permitan evaluar los resultados desde diferentes perspectivas. Para evitar posibles resultados engañosos, o poco fiables, se ha decidido utilizar cuatro métricas diferentes que evalúan de formas diferentes las relaciones entre los candidatos propuestos por los modelos y aquellos considerados como válidos en el conjunto de datos.

Estas métricas permiten evaluar y comparar los modelos en diferentes escenarios, teniendo en cuenta aquellos que devuelven menor número de candidatos respecto a otros. También permiten evaluar, al menos parcialmente, las estrategias de cada modelo para ordenar sus candidatos (SR). Estas métricas, que se describen a continuación, son ACC@1, potential@k, accuracy@n@top1 y MAP@k, donde k $\in$ 3, 5, 10 y n $\in$ 1, 2, 3.

- **Exactitud** (*Accuracy*). Referido en la tarea como *Acc*@1, representa

el ratio de casos en las que el primer sustituto elegido para la palabra objetivo se encuentra en la lista de resultados correctos (*gold standard*), independientemente de su orden. Es la principal métrica utilizada para evaluar el rendimiento de los modelos presentados.

- **Exactitud top** (*Accuracy Top*). Referido en la tarea como $Acc@3@Top1$, $Acc@2@Top1$ y $Acc@3@Top1$, estas tres métricas tratan de aportar más detalle a la medida general de exactitud. Representan el porcentaje de casos en los que al menos uno de los primeros k(1,2 o 3) sustitutos elegidos coinciden con el candidato más frecuente (más utilizado por los anotadores de los datos) de la lista de resultados correctos.

- **Potencial** (*Potential*). Referido en la tarea como $Pot@k$, mide el ratio de casos en el que al menos uno de los primeros k(3,5 o 10) candidatos generados se encuentra en la lista de resultados correctos (*gold standard*). Se divide en $Pot@3$, $Pot@5$ y $Pot@10$.

- **MAP** (*Mean Average Precision*). Referido en la tarea como $MAP@k$, mide cuantos de los candidatos generados están presentes en los resultados correctos (*gold standard*), de forma similar al potencial, pero teniendo en cuenta su posición en la clasificación de candidatos. Se divide en $MAP@3$, $MAP@5$ y $MAP@10$.

3.5. Aproximaciones más relevantes

Un total de 14 equipos presentaron los resultados de sus sistemas de simplificación léxica para los datos de prueba proporcionados, y algunos otros grupos de investigación han presentado sus trabajos de forma posterior sobre estos mismos datos. De estos, solo seis abordan la tarea en español, los cuales se describen en esta sección. Todos ellos afrontan la tarea de forma multilingüe, presentando resultados para los tres idiomas. En la tabla 3.3 se presenta un resumen de los seis modelos, explicando los recursos y técnicas que utiliza cada uno para la tarea, especificando cuales se utilizan para la sub-tarea de generación de sustitutos (SG) y cuales para la evaluación o selección de ellos (SS/SR). Los resultados de los modelos en la tarea se resumen en las tablas A.2, A.3 y A.4, divididas en las distintas métricas utilizadas.

Modelo	Generación sustitutos (SG)	Evaluación de sustitutos (SR)
UniHD	GTP-3 prompts	GTP-3 prompts
GMU-WLV	MLM (RoBERTa-BNE)	MLM probability, word frequency
PresiUniv	MLM (BETO)	Cos-similarity, POS check
UoM&MMU	PromptLS (EASIER)	Fined-tuned Bert model
PolyU-CBS	MLM (BETO)	MLM probability, GPT-2 (BETO), cos-similarity
CENTAL	MLM (RoBERTa)	Simple vote, binary classifier

Tabla 3.3: Recursos y técnicas de modelos para español del TSAR-2022

3.5.1. Aproximación de UniHD

De todas las aproximaciones publicadas dentro de esta tarea, la utilizada por Aumiller y Gertz es seguramente la más interesante, no solo porque obtiene los mejores resultados tanto en inglés como en español, sino también por la originalidad y sencillez técnica de su propuesta respecto al resto (Aumiller y Gertz, 2023). Se trata de un sistema basado en instrucciones, como los descritos en 2.5.3, a través de un modelo GPT parametrizado.

La mayoría de modelos de simplificación léxica en el estado del arte consisten en procesos complejos (*pipelines*) con varios componentes, cada uno de los cuales se basa en distintos conocimientos técnicos y trata de alcanzar todo su potencial mediante combinaciones entre ellos. Sin embargo, esta aproximación presenta una forma de trabajar alternativa, un proceso mucho más sencillo basado en instrucciones y respuestas de GPT-3, que supera a los otros métodos presentados por un amplio margen. Se trata de una propuesta cuyos mejores resultados tienen lugar con el conjunto de datos en lengua inglesa de la tarea compartida, pero que también supera al resto en lengua española. Además, su metodología es fácilmente automatizable para otros casos de simplificación, incluyendo simplificación sintáctica o para la generación de resúmenes, aunque su carácter generalista podría conllevar dificultades para adaptarla a usuarios con necesidades específicas.

Esta aproximación consiste en un conjunto de seis plantillas, con distintos niveles de contexto, mediante las que generar los ≪prompt≫ en inglés para GPT-3. Para aplicarla a otras lenguas, como el español, se aplica una técnica de transferencia lingüística que permite la simplificación en idiomas distintos del inglés. Esto permite obtener resultados de vanguardia con una modificación mínima de las instrucciones originales. Estas plantillas se describen más adelante en esta misma sección, y también puede verse un ejemplo en la tabla 3.5.

Al menos en el contexto de esta tarea, donde se trabaja con palabras

complejas previamente identificadas, el modelo neuronal genera resultados excepcionales. De la misma forma, este método ofrece perspectivas prometedoras para enfoques multilingües y translingüísticos, a tenor de los resultados mostrados por sus autores, que también indican sus posibilidades a la hora de tratar con sistemas con recursos limitados.

Como problema a abordar en el futuro, los autores indican la naturaleza inestable de los LM neuronales, que implican que en casos de entradas similares, la calidad de la respuesta puede variar mucho de una muestra a otra, e incluso el rendimiento puede llegar a fallar por completo. También sugieren el uso de enfoques para generar recursos estáticos a partir de vLLMs (*Very Large Language Models*), que luego pueden utilizarse como datos de entrenamiento para sistemas más baratos, en términos de computación. Por último, consideran que la exploración de enfoques de ajuste para los *prompt* para la búsqueda automatizada de plantillas más adecuadas también aceleraría el proceso de desarrollo de aplicaciones específicas del dominio.

A continuación se explican las dos aproximaciones que los autores han utilizado para evaluar el sistema, así como otros detalles técnicos considerados relevantes.

Primera aproximación: Predicción zero-shot

Dada la complejidad en los diferentes contextos de las palabras a sustituir y con la intención de no sesgar las predicciones del sistema, los autores optaron por implementar esta base de referencia que se basa por completo en una única consulta de "disparo cero". En este caso, al modelo de OpenAI se le proporciona únicamente la frase contextual y la palabra compleja, preguntando al modelo por diez sinónimos simplificados de la palabra compleja en el contexto dado. Esta aproximación no proporciona ningún conocimiento adicional al modelo.

En consecuencia, esta aproximación forma un límite inferior razonable para la configuración de la tarea.

Contexto: [$frase\_de\_contexto$] Pregunta: Question: Given the above context, list ten alternatives for \{$complex\_word$\}" that are easier to understand.

Tabla 3.4: Plantilla de *prompt* para la primera aproximación

Segunda aproximación: Predicciones conjuntas

Existe una gran variabilidad en las generaciones del respuestas por parte del modelo al cambiar la plantilla o la configuración del contexto. Para tratar de moderar este efecto, se utiliza un conjunto de varias plantillas diferentes para ampliar el espectro de posibles generaciones y garantizar que un número mínimo de sugerencias supere el proceso de filtrado.

Las variables en las predicciones pueden agruparse en dos grupos principales, aquellas que aportan la frase contextual al modelo y aquellas que generan los sinónimos únicamente a partir de la palabra compleja. Además, los diferentes *prompts* también contienen entre cero y dos ejemplos extraídos de los datos del prueba del conjunto de datos, incluidos sus resultados esperados. Por último, también se varía la temperatura de generación, ya que un valor más alto aumenta la probabilidad de una predicción más creativa, para tener una lista de respuestas más diversas.

Una vez establecidas las diferentes plantillas de *prompts*, es necesario implementar un sistema para unificar las respuestas. Para ello, Aumiller y Gertz presentan un modelo para filtrar y re-ordenar las palabras propuestas. Para cada uno de las seis *prompts* p, se pide al modelo que sugiera diez expresiones simplificadas alternativas S_p y las ordene con las mismas reglas que el sistema de *prompt* único de la primera aproximación. Con el objetivo de combinar y volver a ordenar las sugerencias s, asignamos una puntuación de combinación V a cada una de las distintas predicciones $s \in U_p S_p$:

$$V(s) = \sum_p max(5{,}5 - 0{,}5 \cdot rank_{S_p}(s), 0) \tag{3.1}$$

donde $rank_{S_p}(s)$ es la posición de la sugerencia s en el ranking resultante del *prompt* p. Si $s \notin S_p$, se establece que $rank_{S_p}(s) = \infty$. Los parámetros de escala se eligen arbitrariamente y pueden ajustarse para tener en cuenta el número previsto de sugerencias por consulta. Los autores también estiman que la mayor mejora de rendimiento se debe simplemente a que se proporcionan más predicciones que por separado, pues los autores filtran las palabras más complejas, provocando que en las aproximaciones sencillas no siempre se mantengan los diez candidatos aportados por el sistema. No obstante, como ganancia secundaria, observan un comportamiento más coherente en las predicciones más altas, lo que mejora especialmente el rendimiento @1 del conjunto.

Los seis *prompts* utilizados por Aumiller y Gertz y descritas en su publicación se presentan en la tabla 3.5.

Traducción de los prompts

Pese a que su trabajo fundamental es con el conjunto de datos en inglés, los autores de UniHD aplican sus experimentos al español y portugués, aprovechando la dimensión multi-lenguaje de la TSAR-2022. Su aproximación es muy sencilla, trasladar las instrucciones al español o al portugués de forma directa, cambiando la pregunta del *prompt* por: "*Given the above context, list ten alternative* ***Spanish*** *words for 'complex_word' that are easier to understand.*". Es decir, únicamente añadiendo el matiz de pedirle al modelo las respuestas en español, en lugar de inglés, explicitándolo en la propia plantilla.

Esto se debe a que, por defecto, las sugerencias devueltas por el modelo suelen estar en el idioma de las *prompts*, inglés en este caso. Al añadir explícitamente el idioma de salida, los autores se aseguran de que las sugerencias del modelo estén en el idioma esperado. Sin embargo, existe la posibilidad de que en este proceso de traducción, interno al modelo GPT, algunas palabras sencillas puedan traducirse como otras más complejas, explicando el menor rendimiento de la propuesta para el resto de idiomas, respecto al inglés.

Filtrar las predicciones

Las respuestas devueltas por los modelos de OpenAI tienen forma de texto libre, generalmente en forma de listas o enumeraciones separadas por comas. Esto implica un paso adicional para convertir la salida de texto a un formato de palabras candidatas estructuradas acorde a la tarea compartida. Como explican Aumiller y Gertz (Aumiller y Gertz, 2023), esta respuesta varía entre los modelos utilizados y no existe un patrón claro en las respuestas, que no se comportan de forma determinista incluso utilizando estructuras de plantillas. Para ello, los autores implementan varios filtros sencillos que aseguren un formato correcto. Además, las sugerencias del modelo se consideran en orden de clasificación, sin utilizar puntuaciones de confianza de predicción ni información similar para volver a clasificar las predicciones de una sola frase.

Contexto: [$frase\_de\_contexto$] Pregunta: Question: Given the above context, list ten alternatives for \{$complex\_word$\}" that are easier to understand.
Prompt 1: Zero-shot con contexto (temperatura=0.3)
Contexto: [$frase\_de\_contexto$] Pregunta: Question: Given the above context, list ten alternatives for \{$complex\_word$\}" that are easier to understand.
Prompt 2: Zero-shot con contexto (temperatura=0.8)
Context: A local witness said a separate group of attackers disguisedin burqas — the head-to-toe robes worn by conservative Afghan women — then tried to storm the compound. Question: Given the above context, list ten alternative words for "disguised" that are easier to understand. Answer:1. concealed 2. dressed 3. hidden 4. camouflaged 5. changed 6. covered 7. masked 8. unrecognizable 9. converted 10.impersonated Contexto: [$frase\_de\_contexto$] Pregunta: Question: Given the above context, list ten alternatives for \{$complex\_word$\}" that are easier to understand.
Prompt 3: Single-shot con contexto
Context: That prompted the military to deploy its largest warship, the BRP Gregorio del Pilar, which was recently acquired from the United States. Question: Given the above context, list ten alternative words for "deploy" that are easier to understand. Answer: 1. send 2. post 3. use 4. position 5. send out 6. employ 7. extend 8. launch 9. let loose 10. organize Context: The daily death toll in Syria has declined as the number of observers has risen, but few experts expect the U.N. plan to succeed in its entirety. Question: Given the above context, list ten alternative words for "observers" that are easier to understand. Answer: 1. watchers 2. spectators 3. audience 4. viewers 5. witnesses 6. patrons 7. followers 8. detectives 9. reporters 10. onlookers Contexto: [$frase\_de\_contexto$] Pregunta: Question: Given the above context, list ten alternatives for \{$complex\_word$\}" that are easier to understand. Respuesta:
Prompt 4: Two-shot con contexto
Give me ten simplified synonyms for the following word: \{$complex\_word$\}"
Prompt 5: Zero-shot sin contexto
Pregunta: Find ten easier words for "compulsory". Respuesta: 1. mandatory 2. required 3. essential 4. forced 5. important 6. necessary 7. obligatory 8. unavoidable 9. binding 10. prescribed Pregunta: Find ten easier words for \{$complex\_word$\}" Respuesta:
Prompt 6: Single-shot sin contexto

Tabla 3.5: Texto utilizado en los 6 prompts

3.5.2. Aproximación de GMU-WL

La participación del equipo GMU-WLV(North et al., 2022) en tarea compartida TSAR consiste en diferentes experimentos con modelos monolingües, en concreto con BERTimbau, ELECTRA, y RoBERTA-large-BNE, basados en el trabajo previo de Ferrés y Saggion(Ferrés y Saggion, 2022). Además, incorporan a sus sistema el uso de la frecuencia Zipf (Zipf, 2016) para la evaluación de palabras candidatas, aunque con resultados poco clarificadores, detectando algunos problemas respecto a la longitud de las palabras. La frecuencia Zipf se calcula a través de la librería de Python *wordfreq* (Speer et al., 2018), y otorga a cada palabra un valor igual al logaritmo en base 10 del número de veces que aparece por cada mil millones de palabras. Así mismo, muestran que el aprendizaje por transferencia habilita el uso de grandes conjuntos de datos preexistentes para tareas PLN con pocos recursos disponibles, como la propia simplificación léxica en español.

De todos los presentados, su mejor sistema obtuvo el 6º puesto en sistemas en español. Se destaca también el correcto funcionamiento en español de RoBERTA-large-BNE, un modelo preentrenado utilizando un corpus masivo de 570GB de textos limpios, que comprende un total de 135 mil millones de palabras extraidas del Archivo Web del Español construido por la Biblioteca Nacional de España entre los años 2009 y 2019. (Gutiérrez-Fandiño et al., 2021)

3.5.3. Aproximación de PresiUniv

Otra de las aproximaciones más relevantes de la tarea para los datos en español es la presentada por PresiUniv(Whistely, Mathias, y Poornima, 2022). Esta funciona a través de tres etapas, de forma similar a otras aproximaciones, una primera fase de generación de candidatos, una fase intermedia donde se selecciona el candidato idóneo a partir de los candidatos, y una fase final de adaptación de la palabra al contexto. Para ello se apoyan en el uso de tres recursos diferentes, los cuales cambian para cada uno de los idiomas de la tarea. Respecto a su aproximación en español, que presenta unos resultados superiores a la mayoría, utiliza tres herramientas; BETO, FastText y Stanford PoS Tagger, para cada una de las tres etapas. BETO (Cañete et al., 2020) es un modelo de lenguaje pre-entrenado a partir del cual se generan los candidatos a sustituir a la palabra clave. Está creado a partir del modelo BERT, entrenado con la técnica Whole Word Masking y

un conjunto de textos en español. FastText (Grave et al., 2018) es una biblioteca ligera, gratuita y de código abierto que genera representaciones de texto, que los autores utilizan para generar vectores de densidad de palabra, que se pueden comparar para evaluar la similaridad entre ellas. Y finalmente, el etiquetador de discursos (*part-of-speech taggers*) Stanford PoS Tagger para transformar el candidato seleccionado acorde al contexto, tal y cómo se describe en la subtarea de adaptación al contexto.

3.5.4. Aproximación de UoM&MMU

Los autores de UoM&MMU(Vásquez-Rodríguez et al., 2022) presentan un modelo, llamado PromptLS, basado en modelos lingüísticos preentrenados sobre los que se utilizan plantillas predefinidas para obtener los sustitutos posibles del término objetivo. Para ello utilizan varios conjuntos de datos de simplificación léxica dependientes del idioma. En el caso del español, el único dataset utilizado es EASIER corpus(Alarcon, Moreno, y Martínez, 2021), una colección de 260 documentos anotados y verificados por lingüistas y expertos, que llegan a formar 5130 instancias.

En el caso concreto del español, las plantillas utilizadas se han generado a partir de siguiente instrucción; «**Un(a)** ***Prompt1 Prompt2*** **de** ***palabra_objetivo*** **es**», donde Prompt1 puede tener los valores {*palabra, sinónimo*} y Prompt2 es igual a {*fácil, simple*}.

Sobre este dataset los autores han implementado una plantilla específica de prompts, que sencillamente se traduce a cada idioma respecto al original en inglés. Esta técnica se combina con diferentes aproximaciones sobre el contexto de la palabra objetivo, incluyendo una opción que no incluye ningún contexto de la frase. Todo esto de aplica después sobre un modelo de lenguaje enmascarado (MLM), en concreto sobre BERTIN(De la Rosa et al., 2022) para los datos en español. Por último, para maximizar la precisión del modelo, se aplica un paso de posprocesamiento que elimina los candidatos inadecuados.

Tras obtener resultados para las diferentes permutaciones posibles en el sistema, los autores concluyen que su modelo produce en algunos casos unos resultados significativamente superiores para el inglés, respecto al español y el portugués, de forma similar a muchas otras aproximaciones multilingüe. Además, los autores subrayan la relevancia de las palabras contextuales alrededor de una palabra compleja para el desarrollo de la tarea, puesto que

en todas las configuraciones el sistema requiere al menos una ventana de 5 palabras (las cinco palabras anteriores y las cinco posteriores) para obtener un buen rendimiento. De la misma forma, el uso de varios candidatos para una palabra compleja en los datos de entrenamiento aumenta el rendimiento final del sistema.

3.5.5. Aproximación de PolyU-CBS

El modelo presentado por PolyU-CVS(Chersoni y Hsu, 2022) para la tarea consta de un enfoque que aborda la tarea en dos pasos. Para la generación de candidatos (SG) de la palabra compleja, el modelo utiliza un modelo lingüístico enmascarado (MLM). Para el dominio del español, este MLM es BETO (Cañete et al., 2020), un modelo lingüístico basado en BERT (descrito en 2.6.4) pero pre-entrenado exclusivamente con datos en español. Este puede modificar el número de candidatos que genera mediante una variable n, que ya incluyen su propio ranking preliminar.

Como segundo paso, para ordenar los candidatos (SR), utilizan un sistema basado en *transformers* que genera tres métricas distintas para cada palabra, utilizando la media más baja entre las tres para realizar la clasificación final. Estas tres métricas son; la probabilidad de las frases mediante modelos auto-regresivos del lenguaje en GPT2 (Cañete et al., 2020), la probabilidad de las frases mediante MLM (Salazar et al., 2019) y la similitud contextualizada en BERT (Devlin et al., 2018). Las herramientas utilizadas, en el caso el español, para calcular dichas métricas son respectivamente GPT2 (entrenado con BETO), BETO y la similitud coseno.

Sin embargo, los experimentos llevados a cabo por los autores con el conjunto de latos del TSAR-2022 indican que la aproximación que mejores resultados ofrece es aquella que únicamente utiliza la similitud coseno para la tarea de ordenar los candidatos (SR). Los autores apuntan que una posibilidad es que esto ocurra debido a la redundancia de utilizar el mismo modelo de lenguaje en ambos pasos de la tarea.

3.5.6. Aproximación de CENTAL

Al igual que otras propuestas de la tarea, la solución presentada por el equipo de CENTAL(Wilkens et al., 2022) se basa en varios modelos basados en BERT, como RoBERTa. Los autores investigan posibles mejoras contextuales y de clasificación de los sustitutos a partir de de estos modelos. Para

obtener posibles candidatos (SG), este modelo utiliza RoBERTa en el caso del español 2.6.4. De todas las posibilidades estudiadas con este modelo, el más efectivo es la denominada Estrategia de Expansión de Consultas (*The Query Expansion* en inglés), utilizando modelos FastTExt para generar los candidatos a partir únicamente de las palabras propuestas. Respecto a la sub-tarea de ordenación de candidatos (SR), destacan dos estrategias utilizadas por los autores; una votación simple de la frecuencia de un candidato en los diferentes modelos BERT y un clasificador binario en inglés aplicado al resto de idiomas. Aunque este segundo método, el clasificador binario, resulta ser más eficaz para la tarea en inglés, la votación simple obtiene mejores resultados tanto en español como en portugués. Esto se debe probablemente a un desajuste a la hora de sustituir las palabras desde el inglés al resto de idiomas.

3.5.7. Análisis de los resultados de TSAR-2022

Las siguientes tablas A.2, A.3 y A.4 recogen los resultados obtenidos por los participantes de la tarea TSAR en español, así como el sistema base TSAR-LSBert (*baseline*) propuesto por los autores de la tarea a partir de una adaptación del MLM LSBert. Los sistemas presentados superan el rendimiento de este sistema base en la mayoría de métricas, demostrando que se ha realizado un avance significativo en la tarea.

Entre todos los resultados, destacan los de UniHD cuyas aproximaciones supera a todas las demás, tanto aquellas basadas en instrucciones, como las basadas en modelos lingüísticos. El uso de aprendizaje a través de instrucciones, y los modelos GPT en concreto, parecen ser una línea de investigación prometedora para la tarea, no solo por su rendimiento excepcionalmente alto, sino también por su facilidad de implementación, pues no requiere grandes conjuntos de datos ni recursos extensos. Esto también es importante a la hora de plantear sistemas multilingües, y muy especialmente para idiomas minoritarios con recursos limitados. Aun así, el uso de múltiples plantillas para incrementar su rendimiento indica un problema de inestabilidad en sus resultados, algo propio de este tipo de modelos, que debe tenerse muy en cuenta para mejorar su rendimiento. Es especialmente relevante su puntuación de 0.94 en Potencial@10, lo que indica que UniHD obtiene al menos un candidato correcto entre todos los que genera para el 94,02 % de los casos.

Además de los modelos basados en instrucciones, destaca también el uso

de modelos lingüísticos enmascarados específicos del español, especialmente BETO y RoBERTa, en combinación con métricas de información sintáctica. Es también significativo que todos ellos son enfoques desarrollados en inglés, que se han adaptado al español únicamente modificando los recursos y modelos del lenguaje por versiones similares en español. Es importante reseñar la popularidad de los modelos enmascarados en la tarea, puesto la tendencia no solo ocurre en español, sino que en inglés 7 de los 11 modelos presentados hacían uso de esta aproximación.

Con respecto a los modelos enmascarados utilizados en la tarea en español, RoBERTa-large-BNE (Gutiérrez-Fandiño et al., 2021) obtiene los mejores resultados, en combinación con un sistema de clasificación basado en las frecuencias de las palabras en el propio modelo.

		Acc@k@Top1		
Aproximación	**ACC@1**	k=1	k=2	k=3
UniHD(Aumiller y Gertz, 2023)	**0.6521**	**0.3505**	**0.5108**	**0.5788**
PresiUniv(Whistely, Mathias, y Poornima, 2022)	0.3695	0.2038	0.2771	0.3288
UoM&MMU(Vásquez-Rodríguez et al., 2022)	0.3668	0.1603	0.2282	0.2690
PolyU-CBS(Chersoni y Hsu, 2022)	0.3586	0.1630	0.2010	0.2364
GMU-WL(North et al., 2022)	0.3532	0.1820	0.2635	0.3288
CENTAL(Wilkens et al., 2022)	0.3097	0.1467	0.2092	0.2391
LSBert-baseline	0.2888	0.0951	0.1440	0.1820

Tabla 3.6: Resultados para Accuracy de las principales aproximaciones

	MAP@k		
Aproximación	k=3	k=5	k=10
UniHD(Aumiller y Gertz, 2023)	**0.4281**	**0.3239**	**0.1967**
GMU-WL(North et al., 2022)	0.2202	0.1664	0.0994
PresiUniv(Whistely, Mathias, y Poornima, 2022)	0.2145	0.1499	0.0832
UoM&MMU(Vásquez-Rodríguez et al., 2022)	0.2128	0.1506	0.0899
PolyU-CBS(Chersoni y Hsu, 2022)	0.2068	0.1456	0.0850
CENTAL(Wilkens et al., 2022)	0.1826	0.1327	0.0779
LSBert-baseline	0.1868	0.1346	0.0795

Tabla 3.7: Resultados para MAP de las principales aproximaciones

3.6. Discusión y propuesta de experimentación

Los resultados obtenidos en la última tarea TSAR evidencian las posibilidades de los nuevos modelos de instrucciones para afrontar varias sub-tareas

	Potencial@k		
Aproximación	k=3	k=5	k=10
UniHD(Aumiller y Gertz, 2023)	**0.8206**	**0.8885**	**0.9402**
GMU-WL(North et al., 2022)	0.5679	0.6793	0.7717
PresiUniv(Whistely, Mathias, y Poornima, 2022)	0.5842	0.6467	0.7255
UoM&MMU(Vásquez-Rodríguez et al., 2022)	0.5326	0.6005	0.6929
PolyU-CBS(Chersoni y Hsu, 2022)	0.5244	0.5978	0.6793
CENTAL(Wilkens et al., 2022)	0.5000	0.5923	0.6358
LSBert-baseline	0.4945	0.6114	0.7472

Tabla 3.8: Resultados para Potencial de las principales aproximaciones

de la simplificación léxica. Dentro del ámbito de la simplificación a partir de palabras complejas previamente etiquetadas, su rendimiento es superior al resto de aproximaciones actuales y su sencillez técnica facilita la experimentación y el desarrollo de nuevos modelos derivados. Sin embargo, el uso de instrucciones generadas manualmente para las tareas automáticas es un campo reciente donde todavía hace falta mucha experimentación, y donde cada día se publican nuevas técnicas y métodos para la generación de instrucciones (*prompt tuning*) (Lester, Al-Rfou, y Constant, 2021), que pueden mejorar el rendimiento de estos modelos. Además, se ha observado que las instrucciones manuales adolecen frecuentemente de un alto grado de inestabilidad (Liu et al., 2023), pero que este problema puede mitigarse a través de varias técnicas. También se echa en falta un análisis más detallado de la influencia de algunos parámetros en el rendimiento final de las tareas, así como las diferencias existentes a la hora de generar las instrucciones en un idioma o en otro. Por último, dada la velocidad a la que se publican nuevos modelos de lenguaje, ya existen nuevas variantes de los modelos GPT cuyo rendimiento es necesario evaluar.

Para ello, nos proponemos abordar la experimentación con el objetivo de identificar posibilidades de mejora en el proceso de simplificación léxica a través de los modelos de instrucciones, específicamente los modelos GPT, a partir del marco teórico descrito en este capítulo. Se pretende examinar la capacidad de los últimos modelos GPT para la tarea mediante distintas configuraciones de predicciones múltiples, como pueden ser zero-shot, one-shot, etc. También se experimentará con los parámetros del modelo, como la temperatura, que controlan el grado de variación del texto, o la longitud de las respuestas.

Después, se experimentará con el proceso de tratamiento de las respuestas para su adaptación al contexto. Se comprobará cómo se puede variar el formato de las respuestas para que sean lo más acorde posible a los requisitos de la tarea, evitando así problemas de rendimiento en esta parte del modelo. Por último, se examinarán en profundidad las respuestas para tratar de detectar los principales errores del modelo para establecer potenciales direcciones de investigación en el futuro.

Capítulo 4

Experimentación y evaluación de los modelos

4.1. Introducción

Este capítulo describe la metodología utilizada para evaluar los modelos propuestos, a la vez que presenta los resultados obtenidos en la evaluación. Durante el proceso de experimentación profundizamos en la capacidad de los modelos GPT-3 para la simplificación léxica en el marco de trabajo de la tarea TSAR, previamente descrita.

Con el fin de explorar el rendimiento de los modelos, se han realizado pruebas con diferentes parámetros y técnicas cuyos resultados podemos comparar directamente para tratar de extraer conclusiones. Existen varias métricas que miden los resultados y que son de interés en este trabajo; la exactitud, el potencial y el valor MAP, descritos previamente en la sección 3.4.

Los experimentos se han clasificado en secciones, que enfrentan distintas estrategias de transferencia de conocimiento (*transfer learning)*, como el tipo de contexto utilizado o el lenguaje de las instrucciones. Además, todos los experimentos se realizan con varios valores del parámetro temperatura (descrito más adelante), cuyo valor altera las probabilidades de las respuestas de los modelos. Finalmente, se analizarán los resultados obtenidos para tener una idea del rendimiento de los modelos y extraer conclusiones sobre cómo abordar la tarea en el futuro.

4.2. OpenAI API

Para realizar los experimentos de este capítulo se han utilizado varios recursos técnicos, entre los que tiene un papel fundamental la API de OpenAI[1], que proporciona acceso a los modelos GPT y a sus parámetros más relevantes. En base a esta plataforma se han desarrollado los diferentes modelos que se presentan más adelante en este capítulo. Para ayudar a la comprensión de los experimentos, esta sección detalla su funcionamiento y el de los diferentes modelos y parámetros utilizados.

4.2.1. Detalles de los parámetros

A continuación, se describen los diferentes parámetros que los modelos GPT permiten modificar y que pueden alterar su rendimiento.

Modelo de lenguaje

La API de OpenAI se basa en un conjunto diverso de modelos personalizados, cada uno desarrollado con un ajuste específico. En el caso de UniHD(Aumiller y Gertz, 2023), el modelo utilizado es "text-davinci-002", que era el último modelo disponible en el momento de su publicación. Diseñado para la serie de modelos GPT-3.5, que se entrenó con una mezcla de texto y código anterior al cuarto trimestre del año 2021. Sin embargo, para la fecha de realización de este trabajo, existen nuevos modelos y versiones que mejoran su rendimiento y que se describen en detalle en la sección 4.4.1. Este modelo se controla mediante el parámetro *engine* y a lo largo de la presente investigación se han utilizados distintos modelos.

Límite de tokens

El parámetro *max_tokens* controla el número máximo de tokens a generar en las respuestas del modelo. Los autores del modelo de UniHD establecen su valor en 256, para asegurar suficiente espacio para las salidas generadas, aunque indican que en la práctica, la mayoría de las terminaciones están muy por debajo de este límite. Durante esta investigación hemos mantenido el mismo valor durante las pruebas, aunque se ha comprobado que las respuestas siempre tienen una longitud mucho menor.

[1] https://openai.com/

Penalizaciones por frecuencia y presencia

Estos dos parámetros (*frequency and presence penalties* en inglés) pueden tener valores comprendidos entre -2,0 y 2,0. En el caso del **penalizador de frecuencia**, los valores positivos penalizan los tokens nuevos en función de si aparecen en el texto hasta el momento, lo que aumenta la probabilidad de que el modelo tienda a abordar temas nuevos en lugar de repetirse. De forma parecida, los valores positivos del **penalizador de presencia** penalizan los nuevos tokens en función de su frecuencia en el texto hasta el momento, lo que disminuye la probabilidad de que el modelo repita textualmente la misma línea. Estos se utilizan para reducir la probabilidad de muestrear secuencias repetitivas de tokens, y funcionan modificando directamente los logits (probabilidades logarítmicas no normalizadas) mediante una contribución aditiva. La documentación de OpenAI[2] señala que los valores razonables de los coeficientes de penalización oscilan entre 0,1 y 1 si el objetivo es reducir un poco las muestras repetitivas. En cambio, si el objetivo es suprimir en gran medida la repetición, entonces se pueden aumentar los coeficientes hasta 2, pero esto puede degradar notablemente la calidad de las muestras. Además, se pueden utilizar valores negativos para aumentar la probabilidad de repetición. En su artículo, UniHD(Aumiller y Gertz, 2023) establecen para su sistema unos valores de *frequency penalty=0,5*, así como *presence penalty=0,3*, con el objetivo de penalizar conjuntamente los tokens presentes y las repeticiones de tokens. Indican también que estos valores están muy por debajo del máximo, ya que los tokens de sub-palabras individuales pueden estar presentes varias veces en múltiples predicciones y ser válidas. Durante el desarrollo de este trabajo esos valores se han mantenido estables.

Temperatura

En los modelos ChatGPT, GPT-3 y GPT-4 el parámetro *temperature* rige la aleatoriedad del sistema y, por tanto, la creatividad de sus respuestas. Se trata de un valor entre 0.0 y 1.0 en donde los valores más altos hacen que la salida sea más aleatoria, mientras que los valores más bajos generan salidas más centradas y deterministas.

En su sistema, UniHD (Aumiller y Gertz, 2023) adoptan un enfoque

[2]https://platform.openai.com/docs/api-reference/parameter-details

distinto al valor por defecto (*temperature=1,0*), variando la temperatura para garantizar un conjunto de resultados más diverso. En este trabajo se han establecido tres valores para este parámetro, 0.3 (bajo) , 0.5 (medio) y 0.7 (alto) para tratar de establecer su influencia sobre el rendimiento de los modelos.

4.3. Implementación

Para el desarrollo de los experimentos de la forma más eficaz y automatizada posible, se ha desarrollado un «pipeline» de varios procesos encadenados que abarcan todo el proceso, desde la generación inicial de las instrucciones hasta la evaluación de los resultados. Todos los procesos han sido implementados expresamente para este trabajo utilizando el lenguaje de programación Python 3, con la excepción del evaluador TSAR, que está desarrollado por los autores de la tarea y cuyo contenido no se ha modificado. A lo largo de esta sección se describen los diferentes pasos que la forman y su función en el proceso de simplificación. La figura 4.1 representa cómo se conectan los diferentes componentes.

- **Generador de instrucciones:** Como primer paso, este proceso transforma todas las frases en instrucciones o *prompts* para poder enviarse al modelo GPT. A partir del fichero de test de la tarea TSAR, que contiene las frases a simplificar, genera un documento con la lista completa de los *prompts* a utilizar.

- **Conector a GPT:** Este programa hace uso de la API de OpenAI para acceder a los modelos GPT. Como primer paso, configura los parámetros del modelo a partir de un fichero externo, y a continuación envía uno a uno todos los *prompts* generados en el paso anterior, recogiendo sus respuestas en un nuevo documento. Se trata del proceso que más tiempo requiere durante los experimentos, al depender su duración del tráfico de los servidores y su velocidad de respuesta.

- **Post procesador de respuestas:** Una vez obtenidas las respuestas del modelo GPT, es necesario un proceso de adaptación al formato definitivo para la tarea de evaluación. Se trata de un proceso complejo, debido a la naturaleza variable de las respuestas en los modelos GPT, que se detalla en la sección 4.3.1.

- **Evaluador TSAR:** Este script de evaluación está desarrollado por los autores de la tarea compartida TSAR y puesto a disposición de los participantes. Utiliza dos ficheros de entrada, el documento de respuestas correctas (*gold standard*) y otro documento de similar formato, pero con las respuestas del modelo a evaluar. A partir de ellos, genera los resultados de todas las métricas de la tarea.

Durante el desarrollo de este conjunto de procesos, la facilidad de parametrización del mismo permite configurar los diferentes experimentos de forma sencilla y únicamente a través de ficheros de configuración externos. Esto permite al código adaptarse a otros experimentos dentro del ámbito de la tarea, disponible para equipos o investigadores externos, aunque debe tenerse en cuenta que la API de OpenAI requiere una clave de pago para poder funcionar.

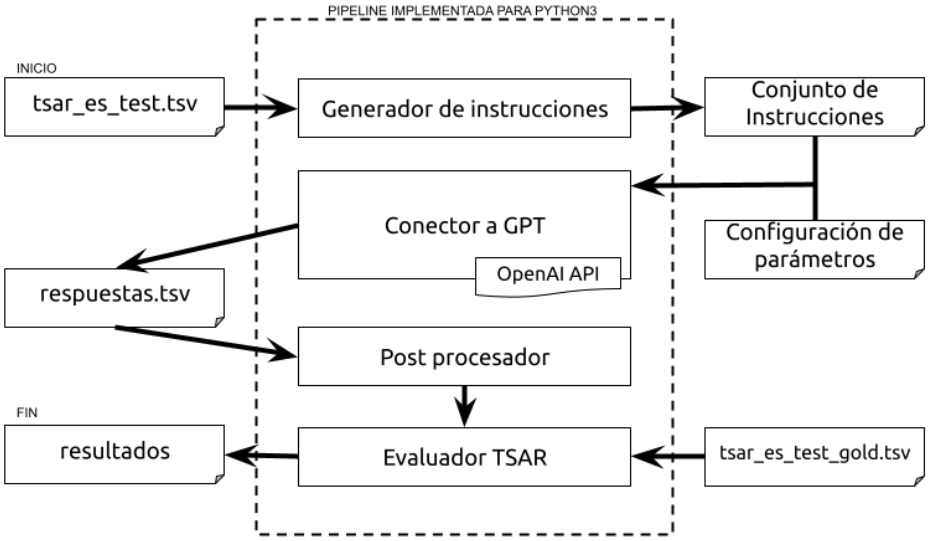

Figura 4.1: Representación del sistema utilizado para los experimentos

4.3.1. Post-procesado de las respuestas

Uno de los problemas de los modelos basados en instrucciones a la hora de automatizar procesos es que estos devuelven sus respuestas de forma libre, con formatos variables, incluso en el caso en que se le entreguen ejemplos previos. Esto implica la necesidad de un paso intermedio que transforme

las respuestas del modelo GPT al formato específico requerido por la tarea TSAR para evaluarlo correctamente.

Coincidiendo con otros trabajos similares (Aumiller y Gertz, 2023), durante este trabajo se ha comprobado que las salidas del modelo GPT no cumplen un patrón estable, incluso aunque se especifique en las instrucciones. Ante este problema de respuestas no deterministas, es necesario este post-procesado para no penalizar su rendimiento. La tabla 4.1 recoge ejemplos de todos los tipos de respuesta que se han encontrado durante la realización de los experimentos.

Ejemplos de respuestas
Answer: 1. declive 2. deterioro 3. desgaste 4. debilitamiento 5. caída 6. descomposición ...
1. Forjarse 2. Ganarse 3. Construirse 4. Crearse 5. Establecerse 6. Lograr ...
Caída, declive, desmejoramiento, desplome, descenso, deterioro, disminución, ...
1. Teniente (lieutenant) 2. Oficial (officer) 3. Subteniente (sub-lieutenant) ...
Despreciando - vilipendiando - denigrando - desacreditando - deshonrando - desdeñando...
1. Conjuntamente; 2. Colectivizado; 3. Cooperativamente; 4. De forma conjunta; ...

Tabla 4.1: Ejemplos de formatos encontrados en las respuestas de los modelos GPT

A continuación, se detallan los filtros que se han implementado durante esta etapa, así como una explicación de los errores que afrontan. Se han desarrollado a partir de los errores encontrados durante la evaluación de las primeras ejecuciones.

- **Numeración de listas:** La mayoría de las respuestas del modelo GPT son en formato de lista numerada. Por ejemplo "*1. adaptarse 2. sumarse 3. incorporarse 4. obedecer 5. ...*". Esto obliga a eliminar la numeración, así como a asegurarse de que todas las palabras se sitúan en la misma línea del documento. Sin embargo, también hay que tener en cuenta que la salida no siempre es así, por lo que primero es necesario identificar si esto ocurre en cada caso.

- **Signos de puntuación:** Fundamentalmente comas (,), puntos y comas (;) y dos puntos (:) que deben eliminarse del texto. Esto se debe a que en ocasiones minoritarias, el modelo GPT enumera las respuestas separadas con comas u otros signos de puntuación.

- **Mayúsculas:** Todas las palabras del conjunto de respuestas de la tarea (*gold-standard*) están en minúsculas, así que es necesario asegurarse de que las respuestas mantienen ese formato.

- **Explicaciones entre paréntesis:** Se ha detectado durante las primeras ejecuciones que en algunas ocasiones, el modelo incluye en su respuesta explicaciones entre paréntesis, formadas por la traducción en inglés. Este contenido adicional no debe ser tenido en cuenta a la hora de generar respuestas válidas para la evaluación.

- **Espacios:** Las palabras candidatas deben separarse mediante tabulaciones, no espacios sencillos, así que también hay que asegurarse de adaptarlo. Esto conlleva la dificultad adicional de tener que respetar los espacios en aquellos sustitutos candidatos formados por dos o más palabras.

4.4. Resultados

Basándose en los trabajos previos de investigación sobre los modelos GPT (Brown et al., 2020; Radford et al., 2019), durante la fase experimental de este trabajo utilizamos el enfoque de ≪transferencia de tareas≫ (*task transfer*, en inglés), donde al modelo se le proporcionan ejemplos de la tarea en el contexto. El uso de estos ejemplos de entrenamiento se describen habitualmente como ≪one-shot≫, en el caso de utilizar un único ejemplo, o ≪few-shot≫ cuando se utilizan varios ejemplos. Estos casos se comparan con la aproximación denominada ≪zero-shot≫, que únicamente utiliza una descripción o invocación de la tarea en lenguaje natural, sin utilizar ejemplos.

Los modelos GPT suelen mejorar su rendimiento con el uso de ejemplos respecto a la falta de ellos (zero-shot), lo que sugiere que los dichos lingüísticos pueden entenderse como meta-aprendices en los que el aprendizaje intrínseco a los LLMs se combina con el aprendizaje rápido ≪en contexto≫ a través de las instrucciones del modelo (Brown et al., 2020).

Para facilitar su comprensión y análisis, los experimentos se han dividido en diferentes secciones que afrontan distintos aspectos. Además de una explicación de los mismos, cada sección incluye una tabla resumen con los resultados obtenidos, a los que se le ha incluido con fines comparativos los resultados de UniHD (Aumiller y Gertz, 2023), por ser el modelo de mejor rendimiento de la tarea TSAR 2022, y del modelo *LSBert-baseline*, utilizado como punto de referencia por los autores de la tarea (Saggion et al., 2023). La totalidad de los resultados se puede encontrar también resumida en el Apéndice A.

4.4.1. Text-DaVinci-003 y GPT-3.5-turbo

Como primera aproximación, se han comparado dos de los modelos de lenguaje más recientes que existen para GPT-3, con la intención de conocer si existe una diferencia relevante entre sus rendimientos, de cara a realizar el resto de los experimentos. También se ha incluido una versión alternativa de uno de los modelos, para definir si la existencia de un mayor número de tokens durante su entrenamiento influye en su rendimiento en la tarea. Los dos modelos escogidos son los siguientes;

- **text-davinci-003:** La última versión del modelo text-davinci-002, utilizado por UniHD(Aumiller y Gertz, 2023) para la tarea. Se ha entrenado en una amplia gama de tareas y tiene una mayor capacidad de aprendizaje a partir de menos ejemplos en comparación con otros modelos.

- **gpt-3.5-turbo:** El modelo GPT-3.5 más reciente. Optimizado para el uso conversacional, también es más preciso en tareas de clasificación y en aprendizaje con contexto. Además, sus respuestas suelen ser más extensas. Se ha incluido también una aproximación de su versión 16k, que tiene las mismas capacidades que el modelo gpt-3.5-turbo estándar pero ha sido entrenado con 4 veces más contexto.

Para esta comparación se ha utilizado una aproximación sencilla de zero-shot en inglés para los tres modelos, variando también los valores de temperatura para obtener un espectro más amplio de resultados. Al evaluar el modelo da-vinci-003 se obtienen unos valores de ACC@1 entre el 61 % y el 67 %, muy similares a los resultados que obtiene UniHD con la versión da-vinci-002. En cambio, al evaluar el modelo gpt-3.5-turbo, obtenemos valores de ACC@1 entre el 68 % y el 72 %, muy superiores a da-vinci y al estado del arte actual. También obtiene unos resultados superiores en MAP@k y en Potencial@k, aunque resulta significativo que la aproximación de UniHD obtiene mejores resultados para algunas secciones de Potencial@k. Respecto a la variable temperatura, ambos modelos obtienen sus mejores resultados de ACC@1 con valores bajos (0.3), aunque en el resto de métricas resultan más dispares. Por su parte, la versión de 16k obtiene peores resultados que su versión estándar. Con todo esto, podemos concluir que el modelo gpt-3.5-turbo tiene un mejor rendimiento que el resto de versiones, lo que resulta

coherente con las recomendaciones de sus autores[3], por tratarse del último modelo publicado. En experimentos posteriores utilizaremos gpt-3.5-turbo en combinación con otras técnicas de *prompting* para mejorar su rendimiento.

			Acc@k@Top1			**MAP@k**			**Potencial@k**		
Aprox.	**Temp.**	**ACC@1**	k=1	k=2	k=3	k=3	k=5	k=10	k=3	k=5	k=10
UniHD (davinci-002)		0.6521	0.3505	0.5108	**0.5788**	0.4281	0.3239	0.1967	0.8206	**0.8885**	**0.9402**
gpt-3.5-t	0.3	**0.7201**	**0.3831**	0.5135	0.5733	**0.5182**	**0.3843**	**0.2243**	**0.8532**	0.8831	0.9021
gpt-3.5-t	0.5	0.6959	0.3804	0.4891	0.5489	0.5055	0.3719	0.2163	0.8233	0.8586	0.8668
gpt-3.5-t	0.7	0.6820	0.3777	**0.5163**	**0.5788**	0.5045	0.3739	0.2170	0.8315	0.8478	0.8722
gpt-3.5-t-16k	0.3	0.6847	0.3722	0.4782	0.5380	0.4987	0.3660	0.2129	0.8070	0.8423	0.8505
davinci-003	0.3	0.6739	0.3586	0.4375	0.5625	0.4038	0.3005	0.1839	0.8369	0.8858	0.9293
davinci-003	0.5	0.6467	0.3369	0.4619	0.5543	0.4364	0.3275	0.1914	0.8260	0.8722	0.9211
davinci-003	0.7	0.6195	0.3125	0.4266	0.5108	0.4101	0.3033	0.1772	0.7853	0.8315	0.8777
LSBert-baseline		0.3262	0.1577	0.2326	0.2860	0.1904	0.1313	0.0775	0.4946	0.5802	0.6737

Tabla 4.2: Resultados de los modelos Text-Da-Vinci-003 y GPT-3.5-turbo.

4.4.2. Idioma de los prompts

Los modelos GPT son multilingües y capaces de gestionar varios idiomas a la vez, pudiendo recibir las instrucciones en un idioma u otro según se le especifique. Esta es una de sus capacidades más útiles a la hora de adaptar tareas a otros idiomas, como ocurre con la tarea TSAR, pero el funcionamiento exacto de esto y cómo influye a su rendimiento específico no se ha evaluado aún. Por esta razón, se ha considerado realizar un experimento alterando únicamente el idioma de los *prompts* para extraer conclusiones de su rendimiento. Aunque GPT-3 acostumbra a responder en el mismo idioma de la instrucción recibida, es posible solicitarle que responda en otro idioma. En el experimento se le han solicitado diez sinónimos en español para la palabra clave, aunque la instrucción se ha redactado en los dos idiomas, tal y cómo se muestra a continuación;

> **Prompt en inglés:**
> Give me ten simplified Spanish synonyms for the following word: "propiciado"
>
> **Prompt en español:**
> Dame diez sinónimos más sencillos en español para la palabra: "propiciado"

Los resultados obtenidos en esta prueba se presentan en la tabla 4.3. Lo primero que podemos observar es que, pese a utilizarse *prompts* sin con-

[3]https://platform.openai.com/docs

texto, los resultados en inglés son similares a los mostrados en la sección anterior, donde se ha utilizado un zero-shot. Este tema se aborda en más detalle a continuación, en la sección 4.4.3. Respecto al lenguaje utilizado en los *prompts*, se observa en los resultados que la versión en inglés obtiene resultados superiores, con ACC@1 en una franja entre 71 % y 73 %, mientras que la versión en español oscila entre 69 % y 71 %. Aunque no es una diferencia elevada, y puede deberse a la variabilidad del modelo, se puede concluir que el uso de instrucciones en español no mejora el rendimiento del modelo, al menos en lo respectivo a esta tarea.

			Acc@k@Top1			**MAP@k**			**Potencial@k**		
Aprox.	**Temp.**	**ACC@1**	k=1	k=2	k=3	k=3	k=5	k=10	k=3	k=5	k=10
UniHD		0.6521	0.3505	0.5108	**0.5788**	0.4281	0.3239	0.1967	0.8206	0.8885	**0.9402**
gpt-3.5-t-NC	0.3	**0.7336**	0.3750	**0.5163**	0.5706	**0.5294**	**0.3916**	**0.2294**	0.8641	**0.8967**	0.9130
gpt-3.5-t-NC	0.5	0.7201	0.3641	0.4972	0.5543	0.5058	0.3743	0.2153	0.8641	**0.8967**	0.9076
gpt-3.5-t-NC	0.7	0.7146	**0.3777**	0.5054	0.5760	0.4992	0.3710	0.2160	0.8668	0.8940	0.9157
gpt-3.5-t-NC-ES	0.3	0.7146	0.3641	**0.5163**	0.5543	0.5123	0.3777	0.2212	**0.8695**	0.8913	0.9048
gpt-3.5-t-NC-ES	0.5	0.6902	0.3559	0.5108	0.5543	0.5033	0.3631	0.2136	0.8505	0.8777	0.8994
gpt-3.5-t-NC-ES	0.7	0.6956	0.3451	0.5000	0.5570	0.5025	0.3765	0.2207	0.8559	0.8940	0.9130
LSBert-baseline		0.3262	0.1577	0.2326	0.2860	0.1904	0.1313	0.0775	0.4946	0.5802	0.6737

Tabla 4.3: Resultados para las aproximaciones con prompts en inglés y español.

4.4.3. Aproximaciones Zero-shot

En las aproximaciones de zero-shot, el modelo no recibe ningún ejemplo en las instrucciones, sino que debe generar su respuesta sin información adicional, aunque sí se comparte la frase contextual como parte de la instrucción, junto con la palabra objetivo. Adicionalmente, se le añade una instrucción sencilla en la que se pide al modelo que aporte diez palabras alternativas para la palabra objetivo. Tampoco se incluyen detalles sobre el formato de salida, ni otras especificaciones. Las respuestas del modelo se consideran en orden de clasificación a efectos de la tarea, y no se utiliza ningún otro proceso para reordenarlas.

			Acc@k@Top1			**MAP@k**			**Potencial@k**		
Aprox.	**Temp.**	**ACC@1**	k=1	k=2	k=3	k=3	k=5	k=10	k=3	k=5	k=10
UniHD		0.6521	0.3505	0.5108	**0.5788**	0.4281	0.3239	0.1967	0.8206	0.8885	**0.9402**
gpt-3.5-t-0s	0.3	0.7201	**0.3831**	0.5135	0.5733	0.5182	0.3843	0.2243	0.8532	0.8831	0.9021
gpt-3.5-t-0s	0.5	0.6959	0.3804	0.4891	0.5489	0.5055	0.3719	0.2163	0.8233	0.8586	0.8668
gpt-3.5-t-0s	0.7	0.6820	0.3777	**0.5163**	**0.5788**	0.5045	0.3739	0.2170	0.8315	0.8478	0.8722
gpt-3.5-t-NC	0.3	**0.7336**	0.3750	**0.5163**	0.5706	**0.5294**	**0.3916**	**0.2294**	0.8641	**0.8967**	0.9130
gpt-3.5-t-NC	0.5	0.7201	0.3641	0.4972	0.5543	0.5058	0.3743	0.2153	0.8641	**0.8967**	0.9076
gpt-3.5-t-NC	0.7	0.7146	0.3777	0.5054	0.5760	0.4992	0.3710	0.2160	**0.8668**	0.8940	0.9157
LSBert-baseline		0.3262	0.1577	0.2326	0.2860	0.1904	0.1313	0.0775	0.4946	0.5802	0.6737

Tabla 4.4: Resultados para las aproximaciones Zero-shot.

No obstante, es importante diferenciar esta aproximación zero-shot, de una aproximación sin contexto (identificada como NC en las tablas de resultados). En el caso de las instrucciones sin contexto, no solo no se aporta un ejemplo de solución al modelo, sino que tampoco se le incluye el contexto de la frase en la que se encuentra la palabra a sustituir. Los resultados obtenidos en este experimento se presentan en la tabla 4.4.

Un ejemplo de aprendizaje cero en nuestro modelo es el siguiente:

Prompt sin contexto:
Give me ten simplified Spanish synonyms for the following word: "acogerse"
Prompt zero-shot con contexto:
Context: *Sufrió una importante reducción en su capacidad para poder acogerse a las normas de la FIFA para los estadios de fútbol.*
Question: Given the above context, list ten alternative Spanish words for *acogerse* that are easier to understand. Answer:

Respuesta de GPT:
1. adaptarse 2. sumarse 3. incorporarse 4. obedecer 5. apegarse 6. ampararse 7. aceptar 8. asimilarse 9. aplicarse 10.aceptarse

4.4.4. Aproximaciones One-shot

De forma similar a las aproximaciones zero-shot, en las aproximaciones one-shot, se solicita al modelo que genere una lista de diez sinónimos de la palabra clave que sean más sencillos. Pero en esta ocasión, al modelo se le aporta un ejemplo de pregunta y respuesta correcta, como parte del *prompt*, previamente a la solicitud para que haga lo mismo con una nueva frase. El ejemplo se ha extraído de las frases de prueba que los autores de la tarea TSAR ofrecen junto al conjunto de datos de evaluación.

			Acc@k@Top1			**MAP@k**			**Potencial@k**		
Aprox.	**Temp.**	**ACC@1**	k=1	k=2	k=3	k=3	k=5	k=10	k=3	k=5	k=10
UniHD		0.6521	0.3505	0.5108	0.5788	0.4281	0.3239	0.1967	0.8206	0.8885	**0.9402**
gpt-3.5-t-1s	0.3	**0.7771**	**0.4266**	**0.5597**	0.6195	**0.5590**	**0.4187**	**0.2471**	**0.8777**	**0.8994**	0.9048
gpt-3.5-t-1s	0.5	0.7635	0.4184	0.5516	**0.6304**	0.5394	0.4022	0.2371	0.8614	0.8967	0.9048
gpt-3.5-t-1s	0.7	0.7527	0.4239	0.5434	0.6141	0.5357	0.4025	0.2358	0.8722	**0.8994**	0.9157
LSBert-baseline		0.3262	0.1577	0.2326	0.2860	0.1904	0.1313	0.0775	0.4946	0.5802	0.6737

Tabla 4.5: Resultados para las aproximaciones One-Shot

Estas aproximaciones mediante ejemplos son extensamente utilizadas en la experimentación con modelos GPT-3, ya que este es capaz de extraer

detalles de una tarea a realizar a través de unos pocos ejemplos expresados en lenguaje natural (Brown et al., 2020). Este proceso sustituye a los procesos clásicos de ajuste fino en los modelos previos, que requerían una gran cantidad de datos. Los resultados de esta aproximación se recogen en la tabla 4.5. Se observa una subida general respecto a las aproximaciones de zero-shot en torno al 3-4 % en todas las métricas, estableciendo además un nuevo máximo de ACC@1 de 77,71 %. De aquí podemos concluir que el modelo está transfiriendo conocimiento (*transfer learning*) a partir del ejemplo incluido en el *prompt*.

Un ejemplo de aprendizaje one-shot en nuestro modelo es el siguiente:

Prompt one-shot:

Context: *Sufrió una importante reducción en su capacidad para poder acogerse a las normas de la FIFA para los estadios de fútbol.*

Question: Given the above context, list ten alternative Spanish words for *acogerse* that are easier to understand. Answer: 1. adaptarse 2. sumarse 3. incorporarse 4. obedecer 5. apegarse 6. ampararse 7. aceptar 8. asimilarse 9. aplicarse 10. aceptarse

Context:*A comienzos de la década de 1980, se trasladó a Los Ángeles, en California, donde comenzó a labrarse una reputación con sus actuaciones, tanto eléctricas como acústicas.*

Question: Given the above context, list ten alternative Spanish words for "labrarse" that are easier to understand. Answer:

Respuesta de GPT:

1. construirse 2. trabajar 3. ganarse 4. formarse 5. cultivar 6. prepararse 7. hacerse 8. trabajarse 9. forjarse 10.crearse

4.4.5. Aproximaciones Two-shots

La configuración two-shots, también llamada few-shots, es similar a los one-shot, con una única excepción de que recibe dos ejemplos previos extraídos de las frases de prueba. Durante este experimento, se han replicado las aproximaciones one-shot, incluyendo un segundo ejemplo extraído del conjunto de datos de prueba de la tarea.Un ejemplo de aprendizaje two-shots en nuestro modelo es el siguiente:

Prompt two-shots:

Context: *Sufrió una importante reducción en su capacidad para poder acogerse a las normas de la FIFA para los estadios de fútbol.*

Question: Given the above context, list ten alternative Spanish words for *acogerse* that are easier to understand. Answer: 1. adaptarse 2. sumarse 3. incorporarse 4. obedecer 5. apegarse 6. ampararse 7. aceptar 8. asimilarse 9. aplicarse 10.aceptarse
Context:*A comienzos de la década de 1980, se trasladó a Los Ángeles, en California, donde comenzó a labrarse una reputación con sus actuaciones, tanto eléctricas como acústicas.*
Question: Given the above context, list ten alternative Spanish words for "labrarse" that are easier to understand. Answer: 1. construirse 2. trabajar 3. ganarse 4. formarse 5. cultivar 6. prepararse 7. hacerse 8. trabajarse 9. forjarse 10.crearse
Context: *El representativo chileno obtuvo una muy buena participación al conquistar los tres primeros lugares del citado certamen.*
Question: Given the above context, list ten alternative Spanish words for "representativo" that are easier to understand. Answer:

Respuesta de GPT:

1. representante 2. característico 3. grupo 4. famoso 5. comisionado 6. simbólico 7. símbolo modelo 8. ejemplar 9. insigne emblemático 10.portavoz

			Acc@k@Top1			**MAP@k**			**Potencial@k**		
Aprox.	**Temp.**	**ACC@1**	k=1	k=2	k=3	k=3	k=5	k=10	k=3	k=5	k=10
UniHD		0.6521	0.3505	0.5108	0.5788	0.4281	0.3239	0.1967	0.8206	0.8885	**0.9402**
gpt-3.5-t-2s	0.3	0.7608	0.4184	0.5489	0.6195	0.5543	**0.4172**	0.2449	0.8722	0.8913	0.9021
gpt-3.5-t-2s	0.5	0.7581	0.4211	**0.5706**	0.6467	0.5455	0.4145	0.2429	0.8804	0.8994	0.9076
gpt-3.5-t-2s	0.7	**0.7717**	**0.4320**	0.5570	**0.6630**	**0.5628**	0.4155	**0.2450**	**0.8940**	**0.9048**	0.9130
LSBert-baseline		0.3262	0.1577	0.2326	0.2860	0.1904	0.1313	0.0775	0.4946	0.5802	0.6737

Tabla 4.6: Resultados para las aproximaciones Two-shots.

Los resultados de este experimento se resumen en la tabla 4.6. En estos se observan pocas diferencias respecto a los modelos one-shot, quedando incluso ligeramente por debajo en la métrica de ACC@1. Esto puede deberse a que el nuevo ejemplo no es relevante para el modelo, pudiendo ser suficiente con uno solo, y las diferencias deberse únicamente a la variabilidad entre ejecuciones. No obstante, para valores altos de k, sus resultados mejoran los del one-shot, lo que puede revelar alguna mejora secundaria en la que el modelo incluye más palabras correctas, aunque esto requeriría más investigación para poder corroborarlo. Puede estar relacionado también con el hecho de que los mejores resultados se obtengan con un valor alto de temperatura, a diferencia del resto de experimentos, pues esto indica una mayor creatividad

en las respuestas del modelo.

4.5. Discusión

En este capítulo se han presentado los distintos experimentos realizados para la simplificación léxica en español mediante modelos de aprendizaje basados en instrucciones. Aquellas aproximaciones con un mayor rendimiento se han recogido en la tabla 4.7, donde se han ordenado según su puntuación de ACC@1, al considerarse la métrica más relevante para la tarea (Saggion et al., 2023). Estos experimentos nos ofrecen una imagen panorámica del funcionamiento de los modelos GPT-3 en el desempeño de la tarea, así como el peso específico de las posibles variaciones en los modelos. De estas pruebas podemos concluir que los nuevos modelos lingüisticos incorporados a GPT-3, especialmente el modelo *gpt-3.5-turbo*, mejora sensiblemente el rendimiento de los anteriores. Nuestras aproximaciones llegan a obtener un 77 % de exactitud (ACC@1) por un 65 % de UniHD(Aumiller y Gertz, 2023), a pesar de no incorporar combinaciones de *prompts*. Este resultado significa que el primer sustituto propuesto por el modelo se encuentra en la lista de resultados correctos en el 77 % de los casos, un resultado significativo y prometedor para el desarrollo de posibles aplicaciones de simplificación de textos. También debemos reseñar las altas puntuaciones que los modelos obtienen en la métrica de potencial, que hace referencia al ratio de candidatos presentes en la lista de resultados correctos. Alcanzando incluso en ocasiones el 90 %, esto representa una capacidad elevada de generar los sustitutos correctos por parte del modelo, dejando la puerta abierta a una importante mejora en su rendimiento si se añade un sistema de ordenación (*ranking*) de candidatos que logre diferenciarlos.

Respecto al papel del parámetro *temperatura*, se observa que los valores más bajos obtienen mejores resultados en la mayoría de los experimentos, pudiendo estimar que un valor más estable de la creatividad del modelo provoca más palabras correctas. Sin embargo, la diferencia es reducida, siempre menor al 5 %, así que su relevancia es menor a otros elementos de la configuración, y esta tendencia se invierte en el caso de las aproximaciones two-shots. También resulta significativo el rendimiento similar de las aproximaciones one-shot y two-shots, que obtienen resultados muy similares en todas las métricas. Esto puede deberse a que el modelo apenas infiere infor-

mación de refuerzo de múltiples contextos, más allá del formato de salida.

Aun así, los resultados obtenidos son susceptibles de mejorar en varias áreas, lo que permite suponer que en el futuro estos modelos lograrán obtener resultados aún más altos. En otras aproximaciones del estado del arte (Chiu, Collins, y Alexander, 2021; Aumiller y Gertz, 2023), la combinación de múltiples *prompts* a través de diversas fórmulas ha generado una mejora significativa en el rendimiento con modelos GPT-3. Las diferencias de resultados en exactitud, MAP y potencial sugieren que los resultados también son susceptibles de mejora cuando se trabaja con listas de varios sustitutos, algo que podría ser útil a la hora de trabajar con entornos de simplificación orientados a usuarios finales con necesidades concretas. Esto es coherente con lo observado durante los experimentos, donde muchas de las respuestas del modelo incluían palabras repetidas, palabras más complejas que el original o combinaciones de dos palabras, para alcanzar los diez sustitutos solicitados. De hecho, algunas aproximaciones como UniHD han optado por borrar estas palabras durante el proceso, aunque eso suponga realizar la evaluación con un menor número de palabras, lo que puede explicar sus resultados con valores altos de k, mucho más cercanos a nuestros experimentos. Además de los problemas con el formato de salida que ya se ha explicado anteriormente, se ha observado otro un error recurrente en los experimentos, consistente en que las palabras candidatas aportadas por el modelo no están conjugadas correctamente respecto a la frase de contexto. Aunque son casos minoritarios, este error invalida la totalidad de las palabras. Una posible solución de futuro para esto sería la inclusión de un etiquetador gramatical que conjugue las palabras erróneas.

			Acc@k@Top1			**MAP@k**			**Potencial@k**		
Aprox.	**Temp.**	**ACC@1**	k=1	k=2	k=3	k=3	k=5	k=10	k=3	k=5	k=10
gpt-3.5-t-1s	0.3	**0.7771**	0.4266	**0.5597**	0.6195	0.5590	**0.4187**	**0.2471**	0.8777	0.8994	0.9048
gpt-3.5-t-2s	0.7	0.7717	**0.4320**	0.5570	**0.6630**	**0.5628**	0.4155	0.2450	**0.8940**	**0.9048**	**0.9130**
gpt-3.5-t-1s	0.5	0.7635	0.4184	0.5516	0.6304	0.5394	0.4022	0.2371	0.8614	0.8967	0.9048
gpt-3.5-t-2s	0.3	0.7608	0.4184	0.5489	0.6195	0.5543	0.4172	0.2449	0.8722	0.8913	0.9021
gpt-3.5-t-2s	0.5	0.7581	0.4211	0.5706	0.6467	0.5455	0.4145	0.2429	0.8804	0.8994	0.9076
gpt-3.5-t-0s	0.3	0.7201	0.3831	0.5135	0.5733	0.5182	0.3843	0.2243	0.8532	0.8831	0.9021
UniHD		0.6521	0.3505	0.5108	0.5788	0.4281	0.3239	0.1967	0.8206	0.8885	0.9402
LSBert-baseline		0.3262	0.1577	0.2326	0.2860	0.1904	0.1313	0.0775	0.4946	0.5802	0.6737

Tabla 4.7: Resultados más significativos ordenados por ACC@1

Capítulo 5

Conclusión y trabajo futuro

Este trabajo ha abordado un caso de estudio sobre la simplificación automática de textos en español, específicamente el problema de sustitución de palabras complejas en textos de carácter general. Durante la revisión del estado del arte, se ha demostrado que las herramientas de aprendizaje profundo (*deep learning*) y de aprendizaje a través de instrucciones (*prompt learning*) superan ampliamente en rendimiento a las aproximaciones anteriores, al tiempo que presentan un potencial de mejora que merece ser tenido en consideración. Además, las capacidades multilingües de algunas de estas herramientas, como GPT-3, permiten alcanzar unos resultados más amplios y relevantes, solucionando así, al menos en parte, la dependencia de recursos específicos de un idioma que afectaba a los modelos previos.

Tras un análisis exhaustivo de las tareas colaborativas existentes, se ha seleccionado la tarea TSAR como marco de referencia para los avances en el caso de estudio, ya que cuenta con el mayor número de propuestas recientes y un conjunto de datos bien desarrollado, aunque con un número limitado de casos anotados. A partir de este marco de referencia, se han realizado experimentos con los modelos actuales de aprendizaje a través de instrucciones, centrandonos especialmente en GPT-3. Utilizando varias técnicas de parametrización y diseño de instrucciones (*prompting*), hemos obtenido un modelo sencillo cuyos resultados superan a los demás modelos actuales, demostrando ser una herramienta valiosa para la tarea de simplificación de textos. Por último, destacar también que la metodología de experimentación utilizada ha sido completamente automatizada, lo que permite el desarrollo de las diversas configuraciones de manera rápida y directa, facilitando trabajos futuros con el modelo.

Los resultados obtenidos indican que este enfoque presenta un método prometedor para la simplificación léxica, especialmente en el contexto de palabras complejas previamente identificadas, donde el modelo muestra un rendimiento elevado a pesar de utilizar datos de entrenamiento etiquetados de forma muy limitada. Estos resultados son coherentes en investigaciones recientes y reflejan las tendencias actuales sobre el rendimiento de los últimos grandes modelos de lenguaje (*LLMs*, por sus siglas en inglés) en diversas tareas de procesamiento del lenguaje natural (PLN). No obstante, es importante tener en cuenta que este enfoque se encuentra dentro de un marco muy específico en el ámbito de las tareas relacionadas con la simplificación de textos, que requiere de otras sub-tareas para ser realmente eficaz.

Como trabajo futuro, este estudio ofrece direcciones prometedoras en varias áreas, tanto para mejorar su rendimiento en la tarea como para extender su aplicación a otros campos de la simplificación. Una vez desarrollado un modelo robusto que sirve cómo base, sería interesante llevar a cabo un estudio sobre el uso de técnicas de combinación de múltiples instrucciones. Esto permitiría estudiar en profundidad las diferencias entre instrucciones y cómo pueden mejorar el rendimiento de la tarea. Además, también sería relevante explorar en mayor profundidad la integración de recursos externos que puedan mejorar el rendimiento, especialmente en lo que respecta a la verificación de respuestas y la selección final de la palabra sustituta entre los diferentes candidatos generados. Finalmente, es necesario afrontar el principal obstáculo de los grandes modelos de lenguaje, su naturaleza inestable que provoca variabilidad en la calidad de las respuestas para una misma instrucción, con su consecuente pérdida de rendimiento.

Como se ha destacado durante este trabajo, la simplificación de textos no se reduce a una única tarea del procesamiento del lenguaje natural (PLN), pues abarca una amplia gama de géneros de textos y requisitos para grupos de usuarios específicos, además de poder enfrentar diferentes niveles, como la simplificación léxica y sintáctica. Los avances discutidos en el presente estudio deberían ser aplicables también al resto de tareas de la simplificación de textos, lo que requerirá futuras investigaciones al respecto, así como conjuntos de datos de evaluación que permitan medir los avances. Las técnicas y los avances más recientes indican que se pueden implementar soluciones altamente eficaces en diversas áreas.

Bibliografía

Bibliografía

[Al-Thanyyan y Azmi2021] Al-Thanyyan, Suha S y Aqil M Azmi. 2021. Automated text simplification: a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

[Alarcón, Moreno, y Martínez2020] Alarcón, Rodrigo, Lourdes Moreno, y Paloma Martínez. 2020. Hulat-alexs cwi task-cwi for language and learning disabilities applied to university educational texts. En *IberLEF@ SEPLN*, páginas 24–30.

[Alarcon, Moreno, y Martínez2021] Alarcon, Rodrigo, Lourdes Moreno, y Paloma Martínez. 2021. Lexical simplification system to improve web accessibility. *IEEE Access*, 9:58755–58767.

[Alarcon, Moreno, y Martínez2023] Alarcon, Rodrigo, Lourdes Moreno, y Paloma Martínez. 2023. Easier corpus: A lexical simplification resource for people with cognitive impairments. *Plos one*, 18(4):e0283622.

[Aluísio y Gasperin2010] Aluísio, Sandra y Caroline Gasperin. 2010. Fostering digital inclusion and accessibility: the porsimples project for simplification of portuguese texts. En *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*, páginas 46–53.

[Anderson, Freebody, y others1981] Anderson, Richard C, Peter Freebody, y others. 1981. Vocabulary knowledge. *Comprehension and teaching: Research reviews*, páginas 77–117.

[Aumiller y Gertz2023] Aumiller, Dennis y Michael Gertz. 2023. Unihd at

tsar-2022 shared task: Is compute all we need for lexical simplification. *arXiv preprint arXiv:2301.01764*.

[Baeza-Yates, Rello, y Dembowski2015] Baeza-Yates, Ricardo, Luz Rello, y Julia Dembowski. 2015. Cassa: A context-aware synonym simplification algorithm. En *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, páginas 1380–1385.

[Beigman Klebanov, Knight, y Marcu2004] Beigman Klebanov, Beata, Kevin Knight, y Daniel Marcu. 2004. Text simplification for information-seeking applications. En *OTM Confederated International Conferences.ºn the Move to Meaningful Internet Systems*

, páginas 735–747. Springer.

[Biran, Brody, y Elhadad2011] Biran, Or, Samuel Brody, y Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. En *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, páginas 496–501.

[Bott et al.2012] Bott, Stefan, Luz Rello, Biljana Drndarević, y Horacio Saggion. 2012. Can spanish be simpler? lexsis: Lexical simplification for spanish. En *Proceedings of COLING 2012*, páginas 357–374.

[Brants y Franz2006] Brants, Thorsten y Alex Franz. 2006. *Web 1T 5-gram corpus version 1.1. Technical report, Google Research.*

[Brown et al.2020] Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, y others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[Cañete et al.2020] Cañete, José, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, y Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. *Pml4dc at iclr*, 2020(2020):1–10.

[Cardon y Grabar2020] Cardon, Rémi y Natalia Grabar. 2020. French biomedical text simplification: When small and precise helps. En *The 28th International Conference on Computational Linguistics*.

[Carroll et al.1998] Carroll, John, Guido Minnen, Yvonne Canning, Siobhan Devlin, y John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. En *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, páginas 7–10. Citeseer.

[Cassotti et al.2022] Cassotti, Pierluigi, Cataldo Musto, Marco DeGemmis, Georgios Lekkas, y Giovanni Semeraro. 2022. swapuniba@ fintoc2022: Fine-tuning pre-trained document image analysis model for title detection on the financial domain. En *Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022*, páginas 95–99.

[Cemri, Çukur, y Koç2022] Cemri, Mert, Tolga Çukur, y Aykut Koç. 2022. Unsupervised simplification of legal texts. *arXiv preprint arXiv:2209.00557*.

[Chen et al.2017] Chen, Ping, John Rochford, David N Kennedy, Soussan Djamasbi, Peter Fay, y Will Scott. 2017. Automatic text simplification for people with intellectual disabilities. En *Artificial Intelligence Science and Technology: Proceedings of the 2016 International Conference (AIST2016)*, páginas 725–731. World Scientific.

[Chersoni y Hsu2022] Chersoni, Emmanuele y Yu-Yin Hsu. 2022. Polyucbs at tsar-2022 shared task: A simple, rank-based method for complex word substitution in two steps. En *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, páginas 225–230.

[Chiu, Collins, y Alexander2021] Chiu, Ke-Li, Annie Collins, y Rohan Alexander. 2021. Detecting hate speech with gpt-3. *arXiv preprint arXiv:2103.12407*.

[Collados2013] Collados, José Camacho. 2013. Splitting complex sentences for natural language processing applications: Building a simplified spanish corpus. *Procedia-Social and Behavioral Sciences*, 95:464–472.

[Crossley, Allen, y McNamara2011] Crossley, Scott A, David B Allen, y Danielle S McNamara. 2011. Text readability and intuitive simplification: A comparison of readability formulas. *Reading in a foreign language*, 23(1):84–101.

[De la Rosa et al.2022] De la Rosa, Javier, Eduardo G Ponferrada, Paulo Villegas, Pablo Gonzalez de Prado Salas, Manu Romero, y Maria Grandury. 2022. Bertin: Efficient pre-training of a spanish language model using perplexity sampling. *arXiv preprint arXiv:2207.06814*.

[Devlin et al.2018] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, y Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Devlin1998] Devlin, Siobhan. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic databases*.

[Ermakova et al.2021] Ermakova, Liana, Patrice Bellot, Pavel Braslavski, Jaap Kamps, Josiane Mothe, Diana Nurbakova, Irina Ovchinnikova, y Eric San-Juan. 2021. Text simplification for scientific information access: Clef 2021 simpletext workshop. En *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, páginas 583–592. Springer.

[Ermakova et al.2022] Ermakova, Liana, Eric Sanjuan, Jaap Kamps, Stéphane Huet, Irina Ovchinnikova, Diana Nurbakova, Sílvia Araújo, Radia Hannachi, Elise Mathurin, y Patrice Bellot. 2022. Overview of the clef 2022 simpletext lab: Automatic simplification of scientific texts. En *International Conference of the Cross-Language Evaluation Forum for European Languages*, páginas 470–494. Springer.

[Evans, Orasan, y Dornescu2014] Evans, Richard, Constantin Orasan, y Iustin Dornescu. 2014. An evaluation of syntactic simplification rules for people with autism. Association for Computational Linguistics.

[Evans2011] Evans, Richard J. 2011. Comparing methods for the syntactic simplification of sentences in information extraction. *Literary and linguistic computing*, 26(4):371–388.

[Ferrés y Saggion2022] Ferrés, Daniel y Horacio Saggion. 2022. Alexsis: a dataset for lexical simplification in spanish. En *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, páginas 3582–3594.

[Gasperin, Maziero, y Aluisio2010] Gasperin, Caroline, Erick Galani Maziero, y Sandra M Aluisio. 2010. Challenging choices for text simplification. En *PROPOR*, páginas 40–50. Springer.

[Glavaš y Štajner2015] Glavaš, Goran y Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? En *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, páginas 63–68.

[Grabar y Saggion2022] Grabar, Natalia y Horacio Saggion. 2022. Evaluation of automatic text simplification: Where are we now, where should we go from here. En *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, páginas 453–463, Avignon, France, 6. ATALA.

[Grave et al.2018] Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, y Tomas Mikolov. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

[Gutiérrez-Fandiño et al.2021] Gutiérrez-Fandiño, Asier, Jordi Armengol-Estapé, Marc Pàmies, Joan Llop-Palao, Joaquin Silveira-Ocampo, Casimiro Pio Carrino, Aitor Gonzalez-Agirre, Carme Armentano-Oller, Carlos Rodriguez-Penagos, y Marta Villegas. 2021. Maria: Spanish language models. *arXiv preprint arXiv:2107.07253*.

[Hasler et al.2017] Hasler, Eva, Adrià de Gispert, Felix Stahlberg, Aurelien Waite, y Bill Byrne. 2017. Source sentence simplification for statistical machine translation. *Computer Speech & Language*, 45:221–235.

[Jiang et al.2020] Jiang, Zhengbao, Frank F Xu, Jun Araki, y Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

[Kandula, Curtis, y Zeng-Treitler2010] Kandula, Sasikiran, Dorothy Curtis, y Qing Zeng-Treitler. 2010. A semantic and syntactic text simplification tool for health content. En *AMIA annual symposium proceedings*, volumen 2010, página 366. American Medical Informatics Association.

[Kang et al.2022a] Kang, Juyeon, Abderrahim Ait Azzi, Sandra Bellato, Blanca Carbajo Coronado, Mahmoud El-Haj, Ismail El Maarouf, Mei

Gan, Ana Gisbert, y Antonio Moreno Sandoval. 2022a. The financial document structure extraction shared task (FinTOC 2022). En *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, páginas 83–88, Marseille, France, Junio. European Language Resources Association.

[Kang et al.2022b] Kang, Juyeon, Abderrahim Ait Azzi, Sandra Bellato, Blanca Carbajo-Coronado, Mahmoud El-Haj, Ismail El Maarouf, Mei Gan, Ana Gisbert, y Antonio Moreno-Sandoval. 2022b. The financial document structure extraction shared task (fintoc 2022). En *Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022*, páginas 83–88.

[Koubaa2023] Koubaa, Anis. 2023. Gpt-4 vs. gpt-3.5: A concise showdown.

[Lastra-Díaz, Lara-Clares, y Garcia-Serrano2022] Lastra-Díaz, Juan J, Alicia Lara-Clares, y Ana Garcia-Serrano. 2022. Hesml: a real-time semantic measures library for the biomedical domain with a reproducible survey. *BMC bioinformatics*, 23(1):23.

[Lester, Al-Rfou, y Constant2021] Lester, Brian, Rami Al-Rfou, y Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

[Li et al.2020] Li, Xiaoya, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, y Jiwei Li. 2020. A unified MRC framework for named entity recognition. En *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, páginas 5849–5859, Online, Julio. Association for Computational Linguistics.

[Liu et al.2023] Liu, Xiao, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, y Jie Tang. 2023. Gpt understands, too. *AI Open*.

[Liu et al.2019] Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, y Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[McEnery, Xiao, y Tono2006] McEnery, Tony, Richard Xiao, y Yukio Tono. 2006. *Corpus-based language studies: An advanced resource book*. Taylor & Francis.

[Miller1995] Miller, George A. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

[Min et al.2020] Min, Bonan, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, y Dan Roth. 2020. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*.

[Moreno-Sandoval, Gisbert, y Montoro2020] Moreno-Sandoval, Antonio, Ana Gisbert, y Helena Montoro. 2020. Fint-esp: A corpus of financial reports in spanish. *Fuster, et al., editors, Multiperspectives in analysis and corpus design*, páginas 89–102.

[Naber2004] Naber, Daniel. 2004. Openthesaurus: Building a thesaurus with a web community. *Retrieved January*, 3:2005.

[Navigli y Ponzetto2010] Navigli, Roberto y Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. En *Proceedings of the 48th annual meeting of the association for computational linguistics*, páginas 216–225.

[Nelken y Shieber2006] Nelken, Rani y Stuart M Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. En *11th Conference of the European Chapter of the Association for Computational Linguistics*, páginas 1611–168.

[North et al.2022] North, Kai, Alphaeus Dmonte, Tharindu Ranasinghe, y Marcos Zampieri. 2022. Gmu-wlv at tsar-2022 shared task: Evaluating lexical simplification models. En *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, páginas 264–270.

[North et al.2023] North, Kai, Tharindu Ranasinghe, Matthew Shardlow, y Marcos Zampieri. 2023. Deep learning approaches to lexical simplification: A survey. *arXiv preprint arXiv:2305.12000*.

[North, Zampieri, y Shardlow2023] North, Kai, Marcos Zampieri, y Matthew Shardlow. 2023. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.

[Nunes et al.2013] Nunes, Bernardo Pereira, Ricardo Kawase, Patrick Siehndel, Marco A Casanova, y Stefan Dietze. 2013. As simple as it gets-a sentence simplifier for different learning levels and contexts. En *2013 IEEE 13th international conference on advanced learning technologies*, páginas 128–132. IEEE.

[Oliveira, Wong, y Hong2010] Oliveira, Francisco, Fai Wong, y Iok-Sai Hong. 2010. Systematic processing of long sentences in rule based portuguese-chinese machine translation. En *Computational Linguistics and Intelligent Text Processing: 11th International Conference, CICLing 2010, Iaşi, Romania, March 21-27, 2010. Proceedings 11*, páginas 417–426. Springer.

[ONU2006] ONU. 2006. Convention on the rights of persons with disabilities (crpd) - article 9 accessibility.

[Ortiz Zambrano et al.2019] Ortiz Zambrano, Jenny, Arturo MontejoRáez, Katty Nancy Lino Castillo, Otto Rodrigo Gonzalez Mendoza, y Belkis Chiquinquirá Cañizales Perdomo. 2019. Vytedu-cw: Difficult words as a barrier in the reading comprehension of university students. En *The International Conference on Advances in Emerging Trends and Technologies*, páginas 167–176. Springer.

[Ortiz-Zambranoa y Montejo-Ráezb2020] Ortiz-Zambranoa, Jenny A y Arturo Montejo-Ráezb. 2020. Overview of alexs 2020: First workshop on lexical analysis at sepln. En *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)*, volumen 2664, páginas 1–6.

[Paetzold y Specia2013] Paetzold, Gustavo y Lucia Specia. 2013. Text simplification as tree transduction. En *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*.

[Paetzold y Specia2015] Paetzold, Gustavo y Lucia Specia. 2015. Lexenstein: A framework for lexical simplification. En *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, páginas 85–90.

[Paetzold y Specia2017a] Paetzold, Gustavo y Lucia Specia. 2017a. Lexical simplification with neural ranking. En *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, páginas 34–40.

[Paetzold y Specia2017b] Paetzold, Gustavo H y Lucia Specia. 2017b. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.

[Paetzold2016] Paetzold, Gustavo Henrique. 2016. *Lexical simplification for non-native english speakers*. Ph.D. tesis, University of Sheffield.

[Powers2020] Powers, David MW. 2020. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

[Qiang et al.2020] Qiang, Jipeng, Yun Li, Yi Zhu, Yunhao Yuan, y Xindong Wu. 2020. Lexical simplification with pretrained encoders. En *Proceedings of the AAAI Conference on Artificial Intelligence*, volumen 34, páginas 8649–8656.

[Radford et al.2019] Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, y others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

[Rello et al.2013] Rello, Luz, Clara Bayarri, Azuki Górriz, Ricardo Baeza-Yates, Saurabh Gupta, Gaurang Kanvinde, Horacio Saggion, Stefan Bott, Roberto Carlini, y Vasile Topac. 2013. Dyswebxia 2.0! more accessible text for people with dyslexia. En *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, páginas 1–2.

[Rico-Sulayes2020] Rico-Sulayes, Antonio. 2020. General lexicon-based complex word identification extended with stem n-grams and morphological engines. En *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020), CEUR-WS, Malaga, Spain*, volumen 23.

[Rojo2016] Rojo, Guillermo. 2016. Corpus textuales del español. En *Enciclopedia de lingüística hispánica*. Routledge, páginas v2–285.

[Saggion et al.2015] Saggion, Horacio, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, y Biljana Drndarevic. 2015. Making it simplext: Implementation and evaluation of a text simplification system for spanish. *ACM Transactions on Accessible Computing (TACCESS)*, 6(4):1–36.

[Saggion et al.2023] Saggion, Horacio, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, y Marcos Zampieri. 2023. Findings of the tsar-2022 shared task on multilingual lexical simplification. *arXiv preprint arXiv:2302.02888*.

[Salazar et al.2019] Salazar, Julian, Davis Liang, Toan Q Nguyen, y Katrin Kirchhoff. 2019. Masked language model scoring. *arXiv preprint arXiv:1910.14659*.

[Schick y Schütze2021] Schick, Timo y Hinrich Schütze. 2021. Generating datasets with pretrained language models. *arXiv preprint arXiv:2104.07540*.

[Shardlow2013] Shardlow, Matthew. 2013. A comparison of techniques to automatically identify complex words. En *51st annual meeting of the association for computational linguistics proceedings of the student research workshop*, páginas 103–109.

[Shardlow2014] Shardlow, Matthew. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

[Shardlow et al.2021] Shardlow, Matthew, Richard Evans, Gustavo Henrique Paetzold, y Marcos Zampieri. 2021. Semeval-2021 task 1: Lexical complexity prediction. *arXiv preprint arXiv:2106.00473*.

[Siddharthan2014] Siddharthan, Advaith. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.

[Speer et al.2018] Speer, Robyn, Joshua Chin, Andrew Lin, Sara Jewett, y Lance Nathan. 2018. Luminosoinsight/wordfreq: v2. 2. *Zenodo [Computer Software]*.

[Štajner2021] Štajner, Sanja. 2021. Automatic text simplification for social good: Progress and challenges. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, páginas 2637–2652.

[Štajner, Drndarevic, y Saggion2013] Štajner, Sanja, Biljana Drndarevic, y Horacio Saggion. 2013. Corpus-based sentence deletion and split decisions for spanish text simplification.

[Stajner et al.2022] Stajner, Sanja, Daniel Ferrés, Matthew Shardlow, Kai North, Marcos Zampieri, y Horacio Saggion. 2022. Lexical simplification benchmarks for english, portuguese, and spanish. *arXiv preprint arXiv:2209.05301*.

[Štajner y Popović2016] Štajner, Sanja y Maja Popović. 2016. Can text simplification help machine translation? En *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, páginas 230–242.

[Tager-Flusberg1981] Tager-Flusberg, Helen. 1981. Sentence comprehension in autistic children. *Applied psycholinguistics*, 2(1):5–24.

[Valcarce et al.2020] Valcarce, Daniel, Alejandro Bellogín, Javier Parapar, y Pablo Castells. 2020. Assessing ranking metrics in top-n recommendation. *Information Retrieval Journal*, 23:411–448.

[Vanderwende et al.2007] Vanderwende, Lucy, Hisami Suzuki, Chris Brockett, y Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618.

[Vásquez-Rodríguez et al.2022] Vásquez-Rodríguez, Laura, Nhung Nguyen, Matthew Shardlow, y Sophia Ananiadou. 2022. Uom&mmu at tsar-2022 shared task: Prompt learning for lexical simplification. En *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, páginas 218–224.

[Vaswani et al.2017] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, y Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

[Vu, Tran, y Pham2014] Vu, Tu Thanh, Giang Binh Tran, y Son Bao Pham. 2014. Learning to simplify children stories with limited data. En *Asian Conference on Intelligent Information and Database Systems*, páginas 31–41. Springer.

[Whistely, Mathias, y Poornima2022] Whistely, Peniel, Sandeep Mathias, y Galiveeti Poornima. 2022. Presiuniv at tsar-2022 shared task: Genera-

tion and ranking of simplification substitutes of complex words in multiple languages. En *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, páginas 213–217.

[Wilkens et al.2022] Wilkens, Rodrigo, David Alfter, Rémi Cardon, Isabelle Gribomont, Adrien Bibal, Watrin Patrick, Marie-Catherine de Marneffe, y Thomas François. 2022. Cental at tsar-2022 shared task: How does context impact bert-generated substitutions for lexical simplification? En *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, páginas 231–238.

[Xue et al.2020] Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, y Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

[Yimam et al.2018] Yimam, Seid Muhie, Chris Biemann, Shervin Malmasi, Gustavo H Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, y Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. *arXiv preprint arXiv:1804.09132*.

[Yimam et al.2017] Yimam, Seid Muhie, Sanja Štajner, Martin Riedl, y Chris Biemann. 2017. Cwig3g2-complex word identification task across three text genres and two user groups. En *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, páginas 401–407.

[Zipf2016] Zipf, George Kingsley. 2016. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books.

[Zotova et al.2020] Zotova, Elena, Montse Cuadros, Naiara Perez, y Aitor García Pablos. 2020. Vicomtech at alexs 2020: Unsupervised complex word identification based on domain frequency. En *IberLEF@ SEPLN*, páginas 7–14.

Apéndice A

Resultados en detalle

Aproximación	**Modelo GPT**	**Temperature**	**Técnica prompt**
Aproximación 1	gpt-3.5-turbo	0.3	zero-shot
Aproximación 2	gpt-3.5-turbo	0.5	zero-shot
Aproximación 3	gpt-3.5-turbo	0.7	zero-shot
Aproximación 4	gpt-3.5-turbo-16k	0.3	zero-shot
Aproximación 5	text-davinci-003	0.3	zero-shot
Aproximación 6	text-davinci-003	0.7	zero-shot
Aproximación 7	text-davinci-003	0.5	zero-shot
Aproximación 8	gpt-3.5-turbo	0.3	one-shot
Aproximación 9	gpt-3.5-turbo	0.5	one-shot
Aproximación 10	gpt-3.5-turbo	0.7	one-shot
Aproximación 11	gpt-3.5-turbo	0.3	zero-shot no context
Aproximación 12	gpt-3.5-turbo	0.5	zero-shot no context
Aproximación 13	gpt-3.5-turbo	0.7	zero-shot no context
Aproximación 14	gpt-3.5-turbo	0.3	two-shots
Aproximación 15	gpt-3.5-turbo	0.5	two-shots
Aproximación 16	gpt-3.5-turbo	0.7	two-shots
Aproximación 17	gpt-3.5-turbo	0.3	zero-shot no context (Español)
Aproximación 18	gpt-3.5-turbo	0.5	zero-shot no context (Español)
Aproximación 19	gpt-3.5-turbo	0.7	zero-shot no context (Español)

Tabla A.1: Configuraciones de los modelos evaluados en este trabajo

		Acc@k@Top1		
Aproximación	**ACC@1**	k=1	k=2	k=3
Aproximación 1	0.7201	0.3831	0.5135	0.5733
Aproximación 2	0.6959	0.3804	0.4891	0.5489
Aproximación 3	0.6820	0.3777	0.5163	0.5788
Aproximación 4	0.6847	0.3722	0.4782	0.5380
Aproximación 5	0.6739	0.3586	0.4375	0.5625
Aproximación 6	0.6195	0.3125	0.4266	0.5108
Aproximación 7	0.6467	0.3369	0.4619	0.5543
Aproximación 8	0.7771	0.4266	0.5597	0.6195
Aproximación 9	0.7635	0.4184	0.5516	0.6304
Aproximación 10	0.7527	0.4239	0.5434	0.6141
Aproximación 11	0.7336	0.375	0.5163	0.5706
Aproximación 12	0.7201	0.3641	0.4972	0.5543
Aproximación 13	0.7146	0.3777	0.5054	0.5760
Aproximación 14	0.7608	0.4184	0.5489	0.6195
Aproximación 15	0.7581	0.4211	0.5706	0.6467
Aproximación 16	0.7717	0.4320	0.5570	0.6630
Aproximación 17	0.7146	0.3641	0.5163	0.5543
Aproximación 18	0.6902	0.3559	0.5108	0.5543
Aproximación 19	0.6956	0.3451	0.5000	0.557

Tabla A.2: Resultados de exactitud de los modelos

	MAP@k		
Aproximación	k=3	k=5	k=10
Aproximación 1	0.5182	0.3843	0.2243
Aproximación 2	0.5055	0.3719	0.2163
Aproximación 3	0.5045	0.3739	0.2170
Aproximación 4	0.4987	0.3660	0.2129
Aproximación 5	0.4038	0.3005	0.1839
Aproximación 6	0.4101	0.3033	0.1772
Aproximación 7	0.4364	0.3275	0.1914
Aproximación 8	0.5590	0.4187	0.2471
Aproximación 9	0.5394	0.4022	0.2371
Aproximación 10	0.5357	0.4025	0.2358
Aproximación 11	0.5294	0.3916	0.2294
Aproximación 12	0.5058	0.3743	0.2153
Aproximación 13	0.4992	0.3710	0.2160
Aproximación 14	0.5543	0.4172	0.2449
Aproximación 15	0.5455	0.4145	0.2429
Aproximación 16	0.5628	0.4155	0.2450
Aproximación 17	0.5123	0.3777	0.2212
Aproximación 18	0.5033	0.3631	0.2136
Aproximación 19	0.5025	0.3765	0.2207

Tabla A.3: Resultados de MAP de los modelos

	Potencial@k		
Aproximación	k=3	k=5	k=10
Aproximación 1	0.8532	0.8831	0.9021
Aproximación 2	0.8233	0.8586	0.8668
Aproximación 3	0.8315	0.8478	0.8722
Aproximación 4	0.8070	0.8423	0.8505
Aproximación 5	0.8369	0. 8858	0.9293
Aproximación 6	0.7853	0.8315	0.8777
Aproximación 7	0.8260	0.8722	0.9211
Aproximación 8	0.8777	0.8994	0.9048
Aproximación 9	0.8614	0.8967	0.9048
Aproximación 10	0.8722	0.8994	0.9157
Aproximación 11	0.8641	0.8967	0.913
Aproximación 12	0.8641	0.8967	0.9076
Aproximación 13	0.8668	0.8940	0.9157
Aproximación 14	0.8722	0.8913	0.9021
Aproximación 15	0.8804	0.8994	0.9076
Aproximación 16	0.8940	0.9048	0.9130
Aproximación 17	0.8695	0.8913	0.9048
Aproximación 18	0.8505	0.8777	0.8994
Aproximación 19	0.8559	0.8940	0.9130

Tabla A.4: Resultados de potencial de los modelos