UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

FACULTAD DE INGENIERÍA - UDELAR

Construcción de herramientas para soporte a la enseñaza de lenguas

Estudiantes:
Bernabé González
Isabel Ivagnes
Joaquín Lejtreger

Tutores:
Luis Chiruzzo
Aiala Rosá

14 de marzo de 2019

Agradecimientos

En primer lugar queremos expresar el más profundo agradecimiento a nuestras familias por el incansable apoyo y cariño, no sólo durante el último año de proyecto sino a lo largo de toda la carrera.
A todos nuestros amigos por siempre darnos para adelante.
Aunque al principio nos parecía un mero trámite o formalidad, no podemos dejar de agradecer a nuestros tutores, Luis Chiruzzo y Aiala Rosá, ya que de verdad se lo merecen por la infinita paciencia y toda la ayuda.
Agradecemos al grupo de Políticas Lingüísticas de ANEP por la buena disposición en todo momento. En especial agradecemos a Valentina Dubini y a Aldo Rodríguez por recibirnos siempre con los brazos abiertos.
A Leticia Arbelo y Analía Chargas por invitarnos a pasar el día en sus escuelas. Y al resto de los maestros por interesarse verdaderamente en nuestro proyecto, probar las aplicaciones y hacer que el trabajo fuera más placentero mediante sus palabras de apoyo.
Al Plan Ceibal por prestarnos el equipamiento necesario para el desarrollo.

Resumen

Muchas de las escuelas rurales de nuestro país cuentan con un único docente para dictar todas las lecciones y algunas incluso tienen uno solo en todo el centro educativo. Además, no se cuenta con buena conectividad a internet por lo que varias se quedan fuera de los programas de enseñanza de inglés a través de videoconferencia que ofrece el Plan Ceibal. Esta situación provoca que la enseñanza del idioma se vea afectada y que muchos de los alumnos terminen educación primaria sin alcanzar los niveles de inglés esperados.

El presente proyecto, realizado en colaboración con el Programa de Políticas Lingüísticas de ANEP, pretende ayudar a los docentes de las escuelas rurales con las lecciones de inglés, para disminuir la problemática planteada y lograr universalizar la enseñanza. Queremos contribuir con los maestros para que creen los andamiajes necesarios para sus estudiantes. Es necesario que los alumnos adquieran bastante conocimiento sobre vocabulario y estructuras lingüísticas del idioma. Una forma para que practiquen y adquieran este conocimiento es mediante la resolución de ejercicios.

A partir de diferentes recursos que provee el Procesamiento de Lenguaje Natural llevamos a cabo la elaboración de una plataforma de generación automática de ejercicios junto con sus correspondientes soluciones. Luego de investigar acerca de las herramientas existentes, sobre el nivel y la índole de las actividades requeridas y la arquitectura posible, obtuvimos como resultado una serie de recursos y materiales que permiten la construcción de ejercicios y una plataforma con la que se pueden realizar distintos tipos de actividades. La plataforma tiene dos interfaces, una para los maestros y otra para los alumnos. La interfaz de los maestros permite generar cuatro tipos diferentes de ejercicios a partir de textos en inglés. Los docentes cuentan con una base precargada de materiales, que incluye textos, vocabulario y definiciones que construimos a lo largo del proyecto. También se permite editar las actividades generadas antes de guardarlas ya que es una herramienta supervisada por humanos. Del lado de los estudiantes, una vez que realizan los ejercicios, los mismos son corregidos automáticamente y se puede visualizar las respuestas en ese mismo momento. Además agregamos una aplicación de escritorio para que puedan trabajar sin conexión a internet.

Finalmente podemos afirmar que el trabajo cumple con los objetivos propuestos al inicio del proyecto y que, a pesar de ciertos problemas referidos a

la falta de recursos informáticos, se logró implementar una plataforma que puede ser utilizada en los salones de clases. De acuerdo a las métricas alcanzadas por diferentes herramientas, detectamos que existen recursos con los que se pueden lograr ejercicios de calidad pero que a medida que se van mejorando demandan mucho procesamiento y capacidad. Por otro lado las encuestas realizadas indicaron que los maestros están interesados en el proyecto, ya que les sería de mucha utilidad una herramienta de este estilo y usarían los ejercicios con sus alumnos.

Palabras Clave: inglés, procesamiento, generación, automático, enseñanza, modelo de lenguaje, word embeddings, ANEP

Índice general

1. Introducción **1**

1.1. Objetivo general 4

1.2. Objetivos específicos 5

1.3. ¿Por qué elegimos el proyecto? 5

1.4. Organización del documento 6

2. Marco teórico **8**

2.1. Procesamiento de Lenguaje Natural 8

2.2. Técnicas y recursos para el procesamiento de textos 12

2.2.1. Distancia de Levenshtein 13

2.2.2. POS tagging 13

2.2.3. WordNet 14

2.2.4. Word Sense Disambiguation 16

2.2.5. Word Embeddings 18

2.2.6. Modelo de lenguaje 20

2.3. Trabajos relacionados 22

2.3.1. Natural Language Processing for Enhancing Teaching and Learning - Diane Litman 23

2.3.2. Language Muse 24

2.3.3. Sistemas CALL 25

2.3.4. Otros trabajos 28

3. Requisitos y Esquema de la solución **29**

3.1. Requisitos relevados 29

3.1.1. Requisitos funcionales 29

3.1.2. Requisitos no funcionales 30

3.2. Esquema general de la solución 32

4. Solución desarrollada **36**

4.1. Recursos generados 36

4.1.1. Tesauro infantil 36

4.1.2. Lista de palabras Starters y Movers 37

4.1.3. Palabras frecuentes 38

4.1.4. Vocabulario 38

4.1.5. Corpus 38

4.1.6. Modelo de lenguaje 40

4.1.7. Modelo Word Embeddings 42

4.2. Ejercicios 44

4.2.1. Ejercicio de conjugación de verbos 45

4.2.2. Ejercicio de sustantivos 47

4.2.3. Ejercicio de uso del lenguaje 49

4.2.4. Ejercicio de hipónimos 56

4.3. Evaluación 60

4.3.1. Modelo de lenguaje 60

4.3.2. POS tagging 62

4.3.3. Ejercicio uso de lenguaje 62

4.3.4. Ejercicio hipónimos 66

5. Aplicación y Arquitectura **68**

5.1. Aplicación 68
5.1.1. Interfaz de docentes 68
5.1.2. Interfaz de alumnos 71
5.2. Arquitectura 75
5.2.1. Arquitectura sin conexión 77
5.3. Distribución de la plataforma 78

6. Actividades de vinculación **80**
6.1. Reuniones con ANEP 80
6.2. Visita a Lavalleja 81
6.3. Foro de lenguas 83
6.4. Encuesta a docentes 85

7. Conclusiones **87**
7.1. Conclusiones 87
7.2. Trabajos a futuro 88

Glosario **93**

A. Procesamiento corpus **96**
A.1. Script procesamiento corpus 96

B. Niveles de inglés **98**

C. Entrevistas a los maestros **102**

Bibliografía **108**

Índice de figuras

2.1. Diagrama de flujo de técnicas del procesamiento de lenguaje natural. 9

2.2. Un ejemplo de dos árboles sintácticos de una oración que se puede interpretar de distintas maneras. 10

2.3. Interfaz gráfica de Language Muse al seleccionar un texto para crear una actividad 25

2.4. Interfaz gráfica de WERTi 27

3.1. Flujo para la generación de un ejercicio. 32

4.1. Probabilidades calculadas de modelo entrenado con corpus de Wikipedia simple 41

4.2. Probabilidades calculadas de modelo entrenado con corpus de Wikipedia 41

4.3. Probabilidades calculadas de modelo reducido con corpus de Wikipedia 42

4.4. Los vectores más similares a la palabra "bear", según los tres modelos entrenados. 42

4.5. El ejercicio de conjugación de verbos desde el punto de vista del docente. 45

4.6. El ejercicio de sustantivos desde el punto de vista del docente. 47

4.7. El ejercicio de uso del lenguaje desde el punto de vista del docente. 50

4.8. Estrategia de filtros para obtener distractores 51

4.9. El ejercicio de hipónimos desde el punto de vista del docente. 57

5.1. Las opciones de la interfaz del docente. 68

5.2. Ventana para que el maestro inserte el texto. 69

5.3. Ejercicios que el docente puede generar. 69

5.4. Lista de textos guardados en el servidor. 70

5.5. El diccionario de la base de datos. 70

5.6. Lista de ejercicios almacenados. 72

5.7. Inicio de la aplicación del alumno. 72

5.8. Selector de opciones en el ejercicio de uso del lenguaje. 73

5.9. Ejercicio de uso del lenguaje corregido. 74

5.10. Ejercicio de sustantivos con respuestas corregidas. 74

5.11. Ejercicio de hipónimos desde el lado del estudiante. 75

5.12. Arquitectura de la aplicación 76

5.13. Arquitectura sin conexión de la aplicación 78

6.1. Escuela rural Nº 25, Paso de los troncos 82

6.2. El equipo durante la charla en el foro de lenguas. 84

6.3. El equipo y los maestros durante la demostración en el foro de lenguas. 84

6.4. Heatmap que muestra los resultados de la encuesta referidos a los ejercicios. 86

B.1. Niveles de inglés dentro del MCER. Foto extraída del material brindado por ANEP. 98

B.2. Descriptores y niveles que se espera obtener en las pruebas de vocabulario, lectura y gramática. [50] 99

B.3. Descriptores y niveles que se espera obtener en una prueba de escritura. [50] 100

B.4. Descriptores y niveles que se espera obtener en una prueba de escritura. Foto extraída del material brindado por ANEP. . . . 101

Índice de cuadros

4.1. Comparación de los diferentes conjuntos de vectores generados para palabras del vocabulario de Movers. 43

4.2. Parámetros para la generación de los Embeddings 44

4.3. Parámetros para el filtro de similitud 52

4.4. Comparación de la probabilidad calculada con modelos de lenguaje para diferentes variantes de una oración 61

4.5. Comparación de los tiempos de respuesta de los pos taggers . 62

4.6. Cantidad de palabras por categoría gramatical a partir del filtro de nivel 63

4.7. Cantidad de distractores para sustantivos 63

4.8. Cantidad de palabras retornadas por el filtro con cota de similitud igual a 0.3 y mínimos a retornar igual a 0 64

4.9. Métricas de tiempos de la generación del ejercicio de uso del lenguaje 66

4.10. Comparación de estrategias para aplicar el algoritmo de hipónimos 67

4.11. Evaluación de distancia entre la palabra y su categoría 67

1. Introducción

En el mundo el inglés es la segunda lengua más hablada. Sus usos son tan frecuentes como variados: se utiliza en ámbitos académicos, en carreras profesionales y hasta en una conversación en un viaje al exterior. Es empleado en infinidad de contextos, tales como el económico, científico y académico, y para aquellos que sepan hablarlo les será más sencillo desarrollarse en el mundo profesional. Por todos estos motivos es de suma importancia que desde la niñez se pueda contar con acceso a su aprendizaje.

La enseñanza de inglés en las escuelas públicas en nuestro país ha ido cobrando notoriedad en los últimos años hasta tal punto que ha logrado convertirse en un tema de agenda recurrente dentro del campo educativo.
Varias de las escuelas rurales cuentan con un solo maestro en el salón de clases que tiene la responsabilidad de enseñar todas las materias a sus alumnos, incluyendo las referidas a las lenguas extranjeras. Hay casos en que el maestro no posee los conocimientos suficientes para hacerlo o directamente no está capacitado para dicha tarea [1]. Esto provoca que muchos niños no puedan recibir clases de inglés. Una de las consecuencias es que los estudiantes no se encuentran preparados para los niveles esperados una vez que concurren a los ciclos posteriores de educación primaria (secundaria y terciaria). En la universidad, por ejemplo, muchas veces se asume que los estudiantes ya manejan el idioma.

Con la intención de empezar a solucionar este problema y expandir de a poco el inglés a todo el ciclo de educación primaria, desde el año 2012 se puso en marcha el programa Ceibal en Inglés del Plan Ceibal [2], que está dirigido a todos los estudiantes de 4º, 5º y 6º año de escuelas públicas y estudiantes de educación secundaria. Este programa permite que los estudiantes aprendan con un docente de la lengua extranjera a través de un sistema de videoconferencia instalado en cada uno de los centros educativos. De esta forma se complementa la modalidad de enseñanza presencial por parte del Programa del Departamento de Segundas Lenguas y Lenguas Extranjeras que brinda la Administración Nacional de Educación Pública (ANEP) con la modalidad remota en tiempo real. Los estudiantes no son los únicos que se benefician de este programa. También está orientado al desarrollo profesional de docentes que aprenden, junto con sus alumnos, del profesor remoto.

También se cuenta con la plataforma CREA 2 [3] que permite a los estudiantes continuar con el aprendizaje desde cualquier dispositivo con acceso a internet. La idea detrás de su creación es ofrecer, de forma sencilla, acceso a estudiantes y docentes a los materiales digitales y compartir el contenido generado por los alumnos para su posterior evaluación y corrección por parte del profesor de inglés remoto. El principal problema que presenta es que funciona solamente con internet y a veces puede demorar en descargar los materiales, haciendo que no estén disponibles cuando son necesarios.

En el año 2015 se logró universalizar la enseñanza de inglés en las escuelas urbanas. Allí, el 95 % de los grupos de 4º, 5º y 6º de educación primaria tiene clases de inglés [4]. Sin embargo, en las escuelas rurales el problema sigue estando presente. A continuación se listan las principales dificultades que presentan las escuelas rurales en la enseñanza de inglés, de las cuales cerca de la mitad no logra acceder a ninguno de los programas para la enseñanza de inglés.

- La escasez de profesores de inglés en las escuelas para dictar clases presenciales conduce a que el propio maestro tenga que impartir dichas lecciones, provocando así descontento entre los maestros. Ese problema, como ya mencionamos anteriormente, constituye uno de los más importantes, especialmente cuando se tiene en cuenta que la gran mayoría de estos maestros no cuenta con una formación académica en pedagogía de una segunda lengua y es bastante común que carezcan de las habilidades de inglés necesarias.

- Los grupos de clase por lo general son multinivel, es decir, incluyen diferentes cursos en el mismo salón y para las mismas lecciones, esto implica un trabajo extra para el maestro, que tiene que pensar actividades que los contemplen a todos, dificultando la enseñanza equitativa.

- Algunas de estas escuelas cuentan con una conectividad a internet lenta o intermitente, y, en algunos casos, puede pasar incluso que no haya por un día entero de clase. Esto provoca que ciertas escuelas se queden fuera del programa de profesores remotos que se ofrece con el Plan Ceibal.

- Como resultados de las últimas evaluaciones de inglés, tenemos que en repetidas ocasiones los estudiantes presentan dificultades en la escritura y la lectura, además del habla y la escucha [5].

- Hay escuelas rurales con un solo docente en todo el centro y otras con una matrícula de menos de cinco estudiantes. Actualmente se cuenta con 1.080 escuelas rurales en el país que atienden a 17.000 niños, de las cuales 600 tienen menos de cinco estudiantes [6] y otras donde el simple hecho de estar ubicadas lejos de las rutas principales entorpece que un maestro pueda ir al centro educativo y retornar en el mismo día.

Por la diversidad de las situaciones no es posible aplicar una única solución.

Durante los años 2017 y 2018 se aplicó por parte de ANEP una metodología de e-coaching o entrenamiento no presencial para las escuelas rurales, donde los funcionarios del sector de Políticas Lingüísticas de ANEP brindaron soporte a los maestros con sus clases, ya sea con materiales, capacitación o preparación de lecciones. Nosotros trabajamos dentro del marco de esa propuesta.

Para el año 2020 el Consejo de Educación Inicial y Primaria prevé poner en marcha nuevos planes de aprendizaje de inglés para incluir a todas las escuelas rurales de todo el país. Uno de estos planes es el programa Inglés sin Límites, una propuesta piloto de enseñanza que será implementada en setenta escuelas rurales situadas al norte del Río Negro [7]. Consiste en ofrecer a los propios maestros de las escuelas capacitación, materiales (como libros y videos didácticos para practicar el habla y pronunciación) e infraestructura para que puedan ellos mismos dictar las clases de inglés. De esta forma no se necesitaría de conexión a internet.

Con todo esto se quiere lograr también que los estudiantes alcancen los niveles de inglés esperados. Estos niveles son los establecidos por el Marco Común Europeo de Referencia para las lenguas (MCER) [8] y son usados como referencia por ANEP. En particular, los que nos interesan son A0, A1 y A2. El nivel A0 coincide con el de los que recién se están acercando al idioma, el A1 se corresponde con los usuarios considerados básicos, es decir, aquellos que tienen la capacidad de comunicarse en situaciones de la vida cotidiana para satisfacer necesidades inmediatas con expresiones de uso frecuente, utilizando un vocabulario básico y elemental, ayudado por un interlocutor que hable lento y pausado. Por último, el nivel A2 incluye además a las personas que son capaces de entender y comunicarse en situaciones simples, como por ejemplo introducirse a sí mismo, presentar a su familia y relatar situaciones del pasado usando vocabulario básico.

Por su parte la Universidad de Cambridge ofrece a los escolares una serie de pruebas divertidas y motivadoras llamadas Young Learners English para el idioma inglés y ANEP es centro autorizado para brindar dichas certificaciones [9]. Son tres diferentes pruebas basadas en actividades y se denominan Starters, Movers y Flyers. A su vez, los niveles de Cambridge se corresponden con los del MCER: Starters se encuentra entre A0 y A1 (vendría a ser como un pre A1), Movers con A1 y Flyers con A2 (todos estos niveles se encuentran detallados en el anexo B).

Muchos de los alumnos en las escuelas rurales se encuentran en el nivel A0 [10] y con este trabajo nuestra intención es colaborar con el objetivo de que puedan avanzar de nivel al finalizar el ciclo escolar, centrándonos en que logren por lo menos los niveles pre A1 y A1 (Starters y Movers). El nivel A2 es el objetivo de egreso para educación primaria propuesto por ANEP,

pero se espera que se cumpla para la totalidad de los alumnos recién en el año 2030, según el estado actual de la enseñanza.

1.1. Objetivo general

Los textos de estudio que sirven para preparar a los estudiantes en la toma de pruebas de inglés son cada vez más complejos. Para lograr entenderlos, los alumnos deben tener bastante conocimiento de vocabulario y de estructuras lingüísticas del idioma inglés.

Por su parte, para poder crear nuevos ejercicios para trabajar en clase es menester que los maestros tengan la noción y la capacidad de identificar las estructuras lingüísticas de cada texto. Además necesitan saber cuáles de esas estructuras no son familiares para los alumnos. Incluso cuando en algunos casos tienen esa capacidad, buscar e identificar a mano todos estos elementos puede llegar a ser una tarea bastante lenta y tediosa, sin contar que luego hay que pensar y generar cada ejercicio. Por eso queremos hacer que el proceso sea más eficiente y dinámico.

Con este proyecto tenemos la intención de ayudar en la enseñanza del idioma, enfocándonos en la situación de las escuelas rurales. Por el lado de los estudiantes, queremos que sean capaces de entender esos textos y, por el lado de los maestros, buscamos que se vea facilitada la tarea de elaboración de material.

Una forma de ayudar es mediante ejercicios que permitan a los alumnos centrarse en los elementos lingüísticos que se encuentran en los textos. Se propone llevar a cabo la construcción de herramientas que den soporte a la enseñanza de lenguas a alumnos escolares mediante el uso de técnicas de Procesamiento de Lenguaje Natural. El proyecto se realiza dentro del marco de una colaboración entre la Facultad de Ingeniería y el Programa de Políticas Lingüísticas de ANEP (cuya sigla corresponde a PPL-ANEP), enfocándonos en los alumnos de escuelas rurales en diferentes departamentos de nuestro país.

Nuestro trabajo trata de elaborar una plataforma que permita la generación automática de ejercicios con sus respuestas precalculadas, de modo que la corrección para los estudiantes también sea automática. Algunos recursos de análisis lingüístico que brinda el Procesamiento de Lenguaje Natural pueden ser de gran utilidad para el análisis de textos y la generación, a partir de ellos, de diferentes tipos de ejercicios.

1.2. Objetivos específicos

En base a las necesidades planteadas por ANEP, teniendo en cuenta nuestros propios intereses y evaluando el alcance para que el proyecto tenga la duración establecida identificamos diferentes objetivos, los cuales se listan a continuación.

- Ayudar a los maestros a hacer que alumnos que tienen poco o nulo conocimiento del inglés puedan adquirir los conocimientos esperados correspondientes a los grados de educación primaria que se encuentren cursando. Se pretende que los estudiantes en el nivel A0 de inglés puedan avanzar a los niveles pre A1 y A1 (principiante avanzado) o a los niveles Starters y Movers que se corresponden con los definidos por la Universidad de Cambridge. Se busca que, al momento de ingresar al ciclo de educación secundaria, los estudiantes lo hagan con uno de estos niveles. De esta forma se pretende colaborar con la causa de universalizar la enseñanza del inglés en el ciclo de enseñanza primaria.
- Lograr implementar una herramienta o plataforma que facilite el trabajo de los maestros de escuelas rurales, permitiéndoles ahorrar tiempo en confección de ejercicios. Además brindarles la posibilidad de contar con nuevas actividades cada vez que quieran trabajar con un texto nuevo.
- Crear una aplicación que sea intuitiva, fácil de usar y que no requiera de una buena conexión a internet.
- Investigar sobre el Procesamiento de Lenguaje Natural aplicado a la educación, aprender sobre qué se implementó hasta el momento y los trabajos existentes, y qué técnicas se pueden usar para facilitar la tarea de enseñanza de idiomas.
- Aplicar métodos del Procesamiento de Lenguaje Natural en la enseñanza de inglés, viendo hasta qué punto pueden llegar a ser viables y confiables para el ser humano.
- Tener una experiencia con usuarios reales, tanto con los maestros como con los estudiantes, y tanto en las etapas de desarrollo de las herramientas como para su validación.

1.3. ¿Por qué elegimos el proyecto?

En 2017, los tres realizamos el curso de Introducción al Procesamiento de Lenguaje Natural, donde nos nació un interés por esta área. Por este motivo, a la hora de elegir un proyecto de grado buscamos uno que tuviera que ver

con la misma. Además, queríamos hacer algo que se pudiera aplicar en la vida cotidiana, así que cuando vimos la propuesta de este proyecto, pensamos que era ideal para nosotros. Sentimos en ese momento que el proyecto atacaba una necesidad real, y el hecho de tener que ir y ver la realidad de las escuelas rurales de nuestro país le agregó el factor social que nos aportaba mucho más en contraste con otros proyectos que nos interesaron en su momento. Sentimos que con nuestro trabajo podemos retribuir a la comunidad con lo que hemos aprendido a lo largo de la carrera en facultad. Asimismo, la simple idea de tener que desarrollar un programa capaz de generar ejercicios automáticamente a partir de un texto nos pareció tan original como desafiante.

1.4. Organización del documento

A continuación explicamos la estructura de lo que resta de este informe.

En el capítulo dos llamado Marco Teórico, exponemos una puesta a punto de otros trabajos relacionados, referidos a la enseñanza de diferentes lenguas mediantes el uso del Procesamiento de Lenguaje Natural. Se ofrece una descripción del Procesamiento de Lenguaje Natural junto con las técnicas que se utilizaron en este proyecto y sus aplicaciones más reconocidas. Contiene también en detalle una explicación de las diferentes tecnologías y recursos empleados para que el lector pueda entender mejor los siguientes capítulos.

Dentro del capítulo tres, Requisitos y Esquema de la solución, listamos los requisitos que se establecieron para este proyecto y sintetizamos el resultado final de nuestra solución.

En el capítulo cuatro, cuyo nombre es Solución desarrollada, describimos el trabajo realizado. Trata sobre las herramientas que se implementaron durante la investigación. Además, se hace hincapié en explicar en detalle cada uno de los ejercicios que se construyeron. También hablamos de los problemas más importantes que surgieron a lo largo de la implementación, junto con las soluciones y posibles mejoras. Al final de este capítulo se encuentran las comparaciones junto con las métricas correspondientes de aplicar diferentes técnicas del Procesamiento de Lenguaje Natural dentro de los ejercicios.

El capítulo cinco, Aplicación y Arquitectura, describe la plataforma construida, su arquitectura e interfaces y las instrucciones básicas para realizar un flujo completo. Finaliza con una breve explicación de los recursos informáticos que fueron necesarios y del proceso que se realizó para lograr la distribución de la plataforma.

El capítulo número seis, Actividades de vinculación, narra las experiencias vividas más significativas con usuarios reales que enriquecieron este trabajo, por parte del equipo junto con los tutores, los funcionarios de ANEP y

principalmente con los maestros de las escuelas. Además este capítulo finaliza con los resultados de una encuesta a maestros sobre el uso de la aplicación y un resumen de las críticas recibidas.

El último capítulo de este documento, denominado Conclusiones, contiene comentarios finales, las conclusiones que obtuvimos y el trabajo a futuro que podría hacerse luego de finalizado el proyecto.

2. Marco teórico

Este capítulo pretende introducir el marco teórico del proyecto con algunos aspectos técnicos del aprendizaje automático y el procesamiento de lenguaje natural que servirán como cimientos para entender mejor las herramientas utilizadas descriptas más adelante en esta sección. Además se procura describir algunos de los principales proyectos o trabajos llevados a cabo que se encuentran vinculados con el tema, con el objetivo de poner en contexto de la situación actual a los lectores. De esta manera se informa sobre los avances y el estado vigente del procesamiento de lenguaje natural dentro de la enseñanza de otros idiomas.

2.1. Procesamiento de Lenguaje Natural

Es un conjunto de métodos y técnicas eficientes desde el punto de vista computacional que conforma una de las ramas de la Inteligencia Artificial (AI). No solo se ocupa de la comprensión del lenguaje natural, si no también de la generación del mismo. Este sistema de procesamiento debe conocer todos los niveles de la lengua (fonética y fonología, morfología, sintaxis y semántica) además de las diferentes convenciones de discurso y de uso (pragmática y discurso) [11].

El procesamiento de lenguaje natural, junto con técnicas de aprendizaje automático, un volumen suficiente de texto e información lingüística en forma de reglas y patrones, permite realizar numerosas actividades relacionadas con el aprendizaje de lenguas. Algunos ejemplos de esto son: analizar oraciones que fueron elaboradas por estudiantes, estudiando sus errores gramaticales y ortográficos, y comprender oraciones para elaborar respuestas a preguntas o mantener una conversación en tiempo real con personas.

La figura 2.1 muestra la secuencia de las actividades más importantes que se pueden realizar. A continuación se describen algunas de ellas.

En primer lugar, está la etapa de preprocesamiento a nivel textual. Aquí, asumimos que un texto es simplemente una sucesión de caracteres. En esta etapa se pueden efectuar las siguientes actividades:

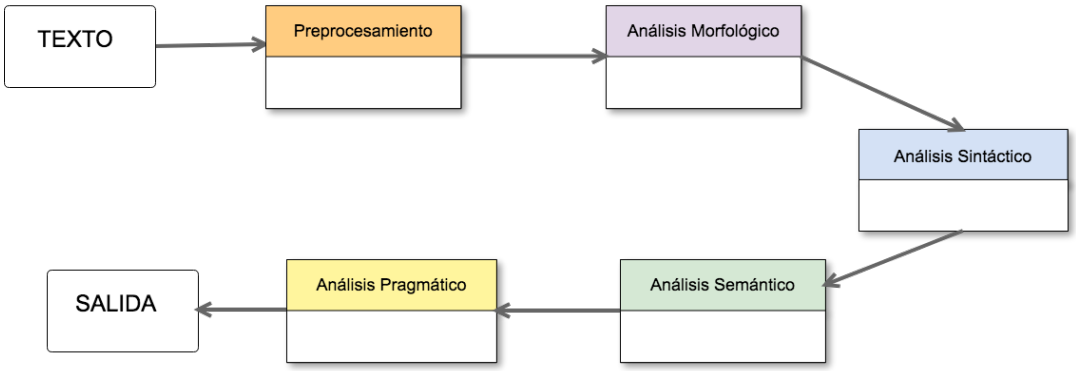

FIGURA 2.1: Diagrama de flujo de técnicas del procesamiento de lenguaje natural.

- Segmentación del texto: como su nombre lo indica, significa dividir el texto en fragmentos tratables para poder trabajar sobre ellos. La dificultad de la tarea en general reside en el formato del texto (plano, marcado, etc.) y lo que se quiera obtener (oraciones, párrafos, etc.). Por ejemplo, para lograr separar en oraciones hay que tener en cuenta las mayúsculas y minúsculas y la puntuación (que, en caso de no tener, la tarea se vuelve más complicada), entre otras cosas. Un signo de puntuación no necesariamente tiene que cumplir una única tarea, el punto puede representar el fin de una oración o formar parte de una sigla o nombre propio.
  Un analizador léxico es el encargado de generar los segmentos, también denominados tokens.

- Filtrado de la información no relevante de los textos: puede ocurrir que junto con los elementos que queremos conservar encontremos otros que deben ser removidos o extraídos para facilitar el tratamiento. Así, si nuestro texto proviene, por ejemplo, de una página web de internet, es probable que contenga diferentes tipos de marcas que definen las características de visualización del sitio, enlaces a otras páginas, objetos no textuales (imágenes, animaciones, tablas, gráficos, etc). Las complicaciones de esta actividad pueden aparecer si el texto está marcado de forma no consistente, dejando entonces de ser una tarea tan sencilla. Otra actividad que podría llevarse a cabo es la de reconocer la lengua en los casos en que no es conocida, ya sea por ejemplo porque internet es la fuente del texto o porque contiene fragmentos correspondientes a lenguas distintas.

- Localización de las unidades básicas a tratar (palabras): Esta tarea es fácil si el espacio o los signos de puntuación actúan como separadores. En las lenguas en que esto no es así, por ejemplo, en el idioma japonés,

o en los casos en que la puntuación no existe, el problema es mayor. Aun habiendo conseguido encontrar las palabras se tiene todavía que lidiar con algunas cuestiones problemáticas. Hay que ser capaces de distinguir los términos multi-palabra, las fechas, los nombres propios, las siglas y abreviaciones, palabras desconocidas o con errores (que no se encuentran en los diccionarios).

La siguiente etapa es la del análisis morfológico, que abarca el etiquetado gramatical de las palabras (denominado POS tagging, del inglés part-of-speech tagging) y la desambiguación de las etiquetas. Esta tarea la lleva a cabo un analizador morfológico que su funcionamiento comúnmente consiste en obtener las unidades significativas más pequeñas que conforman a las palabras o directamente los morfemas. Para llevar a cabo el análisis es necesario contar previamente con los lemas etiquetados con las posibles categorías gramaticales.

El principal problema se presenta cuando queremos etiquetar un lema de forma aislada y tenemos varias categorías admisibles a la vez, entonces el analizador las devolverá todas juntas. Se precisa de conocimiento del mundo para darse cuenta cuál es la categoría correspondiente en un contexto dado. Para un humano es bastante sencillo distinguirlas y asignar la categoría adecuada dado un contexto pero para poder automatizar esta tarea se necesita de desambiguadores morfosintácticos o etiquetadores. En ocasiones los desambiguadores no solucionan del todo el problema y simplemente eliminan las opciones menos probables. Como ocurre en muchas tareas del PLN, los etiquetadores o métodos de etiquetado pueden seguir tres enfoques diferentes, los estadísticos o probabilísticos, los basados en reglas y los híbridos, que son una combinación de los dos primeros.

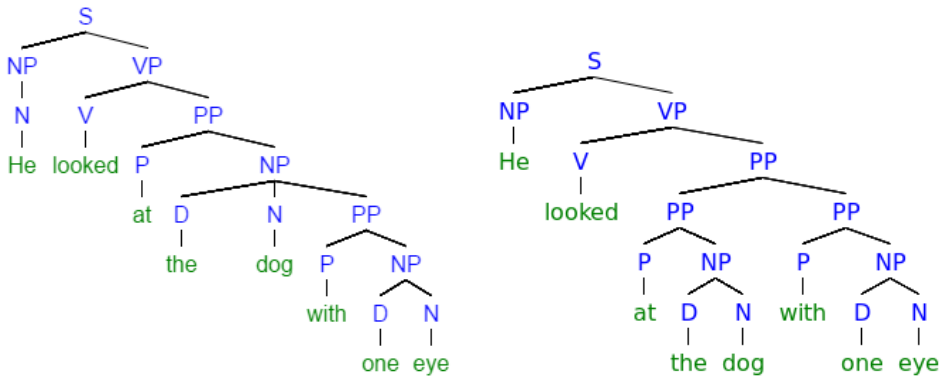

FIGURA 2.2: Un ejemplo de dos árboles sintácticos de una oración que se puede interpretar de distintas maneras.

Para culminar con las etapas que nos interesan queremos hablar del análisis sintáctico o parsing, proceso durante el cual luego del análisis morfológico el texto se puede convertir en otras estructuras de trabajo (por lo general árboles de análisis sintáctico, como se puede ver en la figura 2.2), siempre y

cuando el texto de entrada sea válido. Estas estructuras simplemente son notación de la salida del análisis (también se pueden utilizar paréntesis rectos en lugar de los árboles). Hay ocasiones en que este análisis no es posible o no es conveniente porque el procesamiento es muy lento o costoso. Además, hay casos, como el de la figura, en que la ambigüedad no se puede solucionar.

Por lo general los analizadores que fueron diseñados para la enseñanza de lenguas contienen un componente que prevé o detecta los errores en el caso de que las reglas gramaticales sean violadas. Esta componente, por ejemplo, puede ser lo que se denomina un conjunto de reglas agramaticales (o en inglés buggy rules), que permiten que el parser trabaje sobre oraciones que contienen uno o más errores para luego identificarlos [12].

Existe una infinidad de aplicaciones del procesamiento de lenguaje natural. A continuación listamos algunas a modo de ejemplo.

- Traducción automática: Aunque traducir palabras puede parecer una tarea sencilla, el verdadero desafío reside en entender el significado de las oraciones para proporcionar una traducción más acertada. Otra aplicación previa a la traducción podría ser la de reconocer el idioma del texto si no se conoce con antelación.

- Resumen automático: De gran utilidad cuando necesitamos acceder rápidamente a información específica que se encuentra en un gran conjunto de textos. También esta actividad es importante cuando se emplea para proporcionar una visión general de los textos, mientras se evita la redundancia de múltiples fuentes y se maximiza la diversidad del contenido obtenido.

- Análisis del sentimiento: Es un tipo de análisis semántico que tiene como objetivo identificar la emoción y la opinión que contiene un mensaje incluso cuando no se encuentra totalmente explícito. Muchas empresas utilizan el análisis de sentimiento para saber lo que los usuarios piensan sobre sus productos.

- Recuperación y extracción de la información: Con la gran cantidad de información que existe desde la aparición de internet era necesario contar con técnicas que le puedan mostrar a los usuarios la información que les sea relevante. Por su parte la extracción de información da soporte para luego poder clasificar, resumir y relacionar conceptos.

- Clasificación de textos: Ayuda a categorizar y posteriormente organizar documentos. Un ejemplo bastante común de esto es el filtrado de correos electrónicos como spam o correo basura.

- Respuestas a preguntas: Los sistemas como Siri, Alexa, Google Assistant y asistentes virtuales son cada vez más populares en estos días. Tienen que ser capaces de responder, ya sea de forma escrita o hablada, coherentemente a las preguntas que se les realizan para poder mantener

conversiones fluidas. Aunque se ha avanzando notoriamente en estos sistemas, sobre todo en los últimos años, aún tienen un largo camino para seguir mejorando.

A partir de las aplicaciones mencionadas podemos afirmar que el uso del PLN conlleva a tener numerosos beneficios, ya que ahorra tiempo y agiliza los procesos al automatizar tareas que llevaban a cabo los humanos.

2.2. Técnicas y recursos para el procesamiento de textos

En esta sección se encuentra una explicación detallada sobre las diferentes técnicas y recursos del PLN que decidimos investigar para llevar a cabo la plataforma. Antes de comenzar con cada uno de ellos, proseguimos a contar la razón por la cual decidimos utilizarlos en un principio y por qué elegimos determinados algoritmos.

La distancia de Levenshtein nos pareció un recurso interesante para incluir en el trabajo, no tanto para la generación de los ejercicios sino para mostrar a los estudiantes si estuvieron cerca de la solución correcta en caso de haber errado.

El POS tagging o etiquetado gramatical para el idioma inglés lo necesitamos utilizar para llevar a cabo el procesamiento de todos los textos. Así es como logramos distinguir las palabras que son verbos, sustantivos, preposiciones, etc. Probamos con varios existentes, y al final el que nos dio mejores resultados con respecto a etiquetar adecuadamente según el contexto fue el POS tagger de Stanford.

La base de datos léxica WordNet fue fundamental para encontrar la relación de hiponimia entre palabras, donde una palabra engloba en su significado el de otra. Nos permitió hallar palabras de un texto pertenecientes a ciertas categorías ya establecidas, como fueron colores, animales, comidas, etc.

En algunas ocasiones tuvimos que identificar el significado de una palabra dado un contexto por lo que fue necesario aplicar Word Sense Disambiguation (o desambiguación lingüística), en particular, a través del algoritmo de Lesk. De todos los métodos existentes de Word Sense Disambiguation elegimos utilizar este algoritmo ya que buscábamos una implementación sencilla debido al hecho de que sabíamos de antemano que eran muy pocos los casos en los que se iba a manejar. Esto es por la naturaleza sencilla del inglés que manejamos (niveles pre A1 y A1). El algoritmo de Lesk utiliza WordNet como tesauro o diccionario para distinguir el sentido correcto.

Un modelo Word Embeddings, que en pocas palabras sirve para encontrar relaciones semánticas entre las palabras, nos sirvió para poder identificar palabras que estuvieran dentro de un rango de similitud definido con respecto a otra palabra, es decir, palabras semánticamente similares.

Por último, también generamos un modelo de lenguaje, que usamos con la finalidad de poder deducir la probabilidad que tiene una oración de aparecer con cierta palabra. Para lograr esto utilizamos un modelo de trigramas.

2.2.1. Distancia de Levenshtein

La distancia de Levenshtein (DL) refiere a la distancia mínima de edición entre dos cadenas de caracteres. Las operaciones posibles de edición a nivel de caracter son: inserción, borrado y sustitución.

El algoritmo que se propone para su implementación calcula la cantidad mínima de operaciones necesarias para transformar la cadena de caracteres origen (o) en la cadena de caracteres destino (d) [13]. Por ejemplo:

- Si o es ”manzana” y d es ”manzana”, entonces DL(o,d) = 0, porque no es necesaria ninguna transformación. Ambas cadenas de caracteres son idénticas.
- Si o es ”mono” y d es ”moño”, entonces DL(s,t) = 1, porque una única sustitución (cambiar ”n” por ”ñ”) es suficiente para transformar o en d.

En el modelo original cada operación de edición tiene costo unitario. La distancia de Levenshtein es una forma (no semántica) de indicar qué tan ”similares” o ”cercanas” son las palabras. Su principal aplicación en el área de procesamiento de lenguaje natural es detectar errores ortográficos en las palabras.

Durante el aprendizaje de un nuevo idioma es frecuente escribir una palabra con errores ortográficos. Por lo tanto, utilizamos la distancia de Levenshtein para detectar dichos casos y sugerirle al estudiante la corrección de su error.

2.2.2. POS tagging

POS tagging o etiquetado gramatical es el proceso que dada una secuencia de palabras, etiqueta para cada palabra una categoría gramatical.

Por ejemplo, dada la oración ***”She eats her breakfast”***. El POS tagger luego de analizar la oración nos debería devolver: **[(She, Pronombre), (eats, Verbo), (her, Pronombre), (breakfast, Sustantivo)]**.

Para nuestro proyecto necesitamos un POS tagger para el idioma inglés. Aunque hay POS taggers muy buenos implementados para ese idioma, todos pueden llegar a presentar problemas en presencia de ambigüedad, es decir, cuando una palabra tiene varias categorías gramaticales válidas pero solo una es correcta por el contexto. Decidimos utilizar dos POS taggers, el que provee el toolkit de NLTK [14] y el de Stanford [15].

El POS tagging fue fundamental para detectar las palabras posibles a seleccionar dentro de cada oración. Comenzamos utilizando el POS tagger de NLTK pero encontramos que fallaba en algunas oraciones bastante comunes.

Una oración que ejemplifica esto es ***"The cat jumps"***, donde el POS tagger de NLTK devuelve las siguientes etiquetas gramaticales: **[('The', 'DT'), ('cat', 'NN'), ('jumps', 'NNS')]**
DT corresponde a un determinante, NN un sustantivo y NNS un sustantivo plural. El error es que la palabra **"jumps"** debería tener la etiqueta gramatical de verbo.

Esto sucede porque ninguna implementación de POS tagging es perfecta debido a que se basan en modelos estadísticos y como tales, siempre puede haber casos donde fallan. Sin embargo, se puede mejorar utilizando POS taggers con modelos entrenados a partir de corpus más grandes o con diferentes estrategias.

El POS tagger de Stanford nos arrojó mejores resultados. El ejemplo anterior no falla y devuelve las siguientes etiquetas: **[('The', 'DT'), ('cat', 'NN'), ('jumps', 'VBZ')]**, donde VBZ es un verbo en tercera persona singular.

Por otro lado, el tagger de Stanford es considerablemente más lento que el de NLTK, lo cual tiene un impacto en el rendimiento al generar los ejercicios. En la sección de evaluación (ver 4.3.2) comparamos detalladamente ambos POS taggers.

2.2.3. WordNet

WordNet [16] [17] es una base de datos léxica del idioma inglés. A diferencia de un diccionario, WordNet intenta identificar conceptos y no palabras. Cada concepto o significado se representa con lo que se denomina un synset que incluye un nombre, un número, una pequeña y genérica definición y un identificador que distingue entre sustantivos, adjetivos, verbos y adverbios por sus distintas reglas gramaticales. No incluye en cambio preposiciones, conjunciones, etc.

Por ejemplo, el primer synset para el sustantivo árbol, 'tree.n.01', tiene como definición:

"***a tall perennial woody plant having a main trunk and branches forming a distinct elevated crown; includes both gymnosperms and angiosperms***"

mientras que para el segundo synset 'tree.n.02' su defnición es:

"***a figure that branches from a single root***"

A su vez, el synset 'tree.n.02' tiene asociados dos lemas (tree y tree_diagram), que, para WordNet, implican el mismo concepto.

Gran parte de los synsets están conectados los unos a los otros mediante relaciones semánticas. Para los sustantivos y los verbos encontramos la relación de hiperonimia e hiponimia, asociación subordinada que une términos específicos a términos cada vez más generales. Por ejemplo, **gato** es un hipónimo de **felino** y **felino** es hiperónimo de **gato**. Y en el caso de los verbos, **percibir** es hiperónimo de **escuchar**.

Para los sustantivos también existen las relaciones de meronimia y holonimia, donde la relación indica "ser parte de". Una ventana es un merónimo de una habitación. Las partes se heredan de los niveles superiores: si una silla tiene patas, entonces un sillón también tiene patas. Esto no sucede en sentido contrario ya que las partes pueden ser características de los términos específicos y no ocurrir en las clases más generales, por ejemplo: una silla es un tipo de mueble y tiene patas pero no todos los tipos de muebles tienen patas. Para los adjetivos encontramos que se organizan a través de antonimia, que declara dos conceptos opuestos entre sí, por ejemplo lindo y feo. Éstas son solo algunas de las principales relaciones que existen.

WordNet distingue entre lo que denomina tipos (sustantivos comunes) e instancias (personas específicas, países y entidades geográficas). Las instancias son siempre nodos de las hojas en sus jerarquías.

Podemos contar con una gran cantidad de oraciones dentro de un mismo corpus, lo que implica que las palabras tendrán diferentes significados según el contexto utilizado. Por lo tanto, en lugar de tener una definición fija para cada palabra se definirán en función de su contexto. Debido a que WordNet sólo brinda los sentidos de las palabras pero sin ubicarlos en ningún contexto es complicado utilizarlo sin la ayuda de otro algoritmo para reconocer cuál es el correcto.

Hipónimos e hiperónimos

Queremos hacer énfasis en esta relación ya que fue la que usamos a la hora de crear uno de los ejercicios. Un hipónimo refiere a un término que posee todos los atributos semánticos, o semas, de otro más general (denominado hiperónimo) pero que en su definición incluye además otras características

semánticas que lo diferencian de éste. Su definición contiene también el significado del hiperónimo. Dicho en otras palabras, ”ser hipónimo de” es una relación entre palabras que implica ser un miembro de un grupo semántico. Por ejemplo, ”caballo” es hipónimo de ”animal”, ”auto” es hipónimo de ”vehículo”. La relación de hiponimia es transitiva: si un sillón es un tipo de silla, y si una silla es un tipo de mueble, entonces un sillón es un tipo de mueble.

Una ventaja que tiene WordNet es que los sustantivos están organizados en forma de árbol, de tal manera que el nodo padre de cada uno es su hipónimo. Esto nos permite hallar hipónimos de ciertas palabras en un texto. Para lograr esta estructura de árbol, todos los synsets sustantivos son hipónimos (no necesariamente directos) del synset de ”entity”, pero esto no es verdad para los verbos, cuya jerarquía no está unificada.

2.2.4. Word Sense Disambiguation

En varios idiomas ocurre que una misma palabra puede tener muchos significados, que pueden estar relacionados o no, es decir, que pueden compartir el mismo origen o no compartirlo. A este fenómeno se le denomina homonimia y polisemia, dependiendo de la ocasión. Cuando los significados se encuentran relacionados, se le denomina polisemia. Por ejemplo, la palabra ”sierra” es tanto una herramienta como una conjunto de montañas. En el caso contrario donde no están relacionados y las palabras solo comparten la misma forma, ya sea la pronunciación (palabras homófonas) o la escritura (palabras homógrafas) es cuando se le conoce como homonimia. Como ejemplo tenemos ”ola” y ”hola” que se pronuncian igual en el idioma español pero son dos palabras completamente diferentes con significados y orígenes distintos. Sin conocer la etimología de las palabras no es posible distinguir la homonimia de la polisemia.

Al problema de identificar de qué concepto se está hablando cuando aparece una palabra en un texto se le llama Word Sense Disambiguation (a partir de ahora WSD). Supongamos por ejemplo que se tienen dos de los significados que existen para la palabra ”vela”:

*“**Cilindro de cera o sebo, atravesado por una mecha que se prende para alumbrar.**”*

*“**Conjunto o unión de paños o piezas de lona o lienzo fuerte, que, cortados de diversos modos y cosidos, se amarran a las vergas para recibir el viento que impele la nave.**”*

y las siguientes dos oraciones:

*“**Puso dos velas sobre la mesa.**”*

*"**Es un fabricante de barcos de vela.**"*

Para un humano es fácil darse cuenta que la primera oración utiliza el primer significado y que la segunda está siendo empleada con el segundo. El desarrollo de algoritmos para reproducir esta tarea de forma automática a veces puede ser bastante complejo. Hay diferentes métodos para resolver este problema. Los mismos se pueden encontrar situados en uno de los dos enfoques, los basados en reglas (definidas y escritas por las personas) y los estadísticos (la gran parte del conocimiento se alcanza gracias a los propios datos). Algunos ejemplos son listas de decisión, métodos con Tesauros, Naïve Bayes, etc. En nuestro proyecto decidimos utilizar el algoritmo de Lesk.

Algoritmo de Lesk

El algoritmo de Lesk es uno de los que pertenecen a los métodos con Tesauros y podemos decir que es el más simple. A partir de un tesauro, en nuestro caso WordNet, nos basamos en el supuesto de que dos palabras son similares si ambas comparten palabras en su definición (o en sus "ancestros" de WordNet). Lo que hace el algoritmo de Lesk entonces es devolver, entre todos los synsets posibles, el que tiene más palabras en común con el contexto en su definición.

A continuación se encuentra un pseudocódigo de lo que podría ser su implementación.

Input: Palabra p, frase_contexto s
Output: El synset más probable
$best - sense \leftarrow synset\ mas\ comun$
$contexto \leftarrow palabras\ de\ s$
$mayor_similaridad \leftarrow 0$ **for** $sense \in synsets(word)$ **do**
  $definicion \leftarrow palabras\ en\ la\ definicion$
  $similaridad \leftarrow cantidad_palabras_iguales(definicion, contexto)$
  if $similaridad > maxima_similaridad$ **then**
    z $maxima_similaridad \leftarrow similaridad$
    $best - sense \leftarrow sense$
  end
  return *best-sense*
end

Es importante remarcar que este algoritmo no tiene un muy buen rendimiento en general. El enfoque de Lesk está sujeto a las palabras que conforman las definiciones, por lo que si alguna palabra muy relacionada con el concepto no se encuentra presente en una definición puede cambiar radicalmente los resultados. El contexto de un synset no tiene por qué tener que ver con la definición, y muchas veces los synsets tienen palabras parecidas entre ellos. Por último, el algoritmo solo tiene en cuenta los significados que se encuentran a mano. Esta es una limitación importante, ya que las glosas de

diccionarios tienden a ser bastante cortas y no logran brindar un vocabulario suficiente para relacionar las distinciones de los sentidos.

Sin embargo, gracias a la simplicidad del vocabulario de los textos que utilizamos en nuestro proyecto el algoritmo funciona bien para los ejercicios. Usualmente el synset más común o el primero de las palabras es el que estamos buscando, y para la mayoría de los casos es el que nuestro algoritmo encuentra.

2.2.5. Word Embeddings

Las palabras se pueden pensar inmersas en un espacio vectorial de dimensión finita, donde cada palabra del vocabulario representa un elemento de la base del espacio. Esta abstracción no tiene ninguna utilidad por sí misma, sin embargo, si por medio de una función las palabras se mapean a un espacio vectorial de dimensión menor, se podrían encontrar relaciones interesantes entre las palabras así como de captar el contexto de una palabra en un texto dado.

Así surge la técnica de word embeddings, conjunto de técnicas de modelado de lenguaje y métodos de aprendizaje, que consiste en asignar un vector de un espacio vectorial de dimensión alta a cada palabra, para luego encontrar relaciones semánticas a partir de propiedades de los vectores en el espacio vectorial. La dimensión del espacio, aunque alta, siempre es mucho más pequeña que la cantidad de palabras en el vocabulario, esto es para que los vectores se vuelvan densos, de forma que al seleccionar uno sea similar a los vectores de palabras que aparecen en un contexto parecido. Estas representaciones permiten inferir relaciones semánticas y sintácticas entre las palabras. En matemática, a una función que va desde un espacio vectorial de dimensión alta a uno de dimensión más baja se llama inmersión (embedding).

En pocas palabras esta técnica sirve para representar el vocabulario de un lenguaje y su uso ha demostrado que mejora el rendimiento de algunas tareas como el análisis sintáctico y semántico, parsing, traducción, etc. Se pueden definir funciones algebraicas sobre los vectores que luego pueden ser mapeadas a relaciones gramaticales o semánticas entre las diferentes palabras.

Por último queremos aclarar que los vectores generados dependen del contexto y del corpus de texto que se esté utilizando para describir las palabras por lo que en general no es posible reutilizarlos para la mayoría de los casos.

La implementación que utilizamos de word embeddings se llama word2vec [18] [19], una de las más populares desarrollada por Tomas Mikolov en 2013 en Google. Varios casos de estudio indican que word2vec sirve para identificar relaciones semánticas entre palabras cuyos vectores son cercanos, sin

embargo no es posible tener identificadas todas las relaciones que se encuentran en los vectores.

Funcionamiento

Para generar los vectores, el algoritmo de word2vec está inspirado en una red neuronal, con una única capa oculta de tamaño VxD, donde V es la cantidad de palabras en el vocabulario y D es la dimensión de los embeddings. Dada una oración a la que le falta una palabra, la red neuronal aproxima la probabilidad de que cada palabra "quede bien" en el lugar que falta. Para esto se utiliza un corpus de un tamaño considerable, y se entrenan los pesos de la red viendo cómo aparecen las palabras en los distintos contextos.
Los vectores de palabras, entonces, pasan a ser los pesos de la capa oculta que corresponden a dicha palabra. De esta forma, palabras que aparezcan en contextos parecidos tendrán vectores cercanos (y lo mismo se espera con palabras que sean parecidas semánticamente y aparecen en contextos similares). Para realizar esto se puede utilizar cualquiera de las dos arquitecturas Skip-Gram o Continuous Bag of Words (CBOW), ambas diseñadas como redes neuronales. Skip-Gram funciona correctamente con un conjunto reducido de datos y se le da bien representar palabras que son consideradas raras o poco frecuentes en un vocabulario.

Skip-Gram

Dado un conjunto de palabras $w_1, w_2, ..., w_T$, y una ventana de tamaño c, el modelo Skip-Gram busca maximizar

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq c, j\neq 0} log\ P(w_{t+j}|w_t)$$

Este valor es el promedio de los logaritmos de las probabilidades de que palabras con vectores similares estén en contextos afines.
Hay diferentes formas de aproximar esta probabilidad. [19]

En el caso de CBOW, predecimos la palabra central en función de las palabras de un contexto dado y en el caso de Skip-gram, es lo opuesto. El nombre de Bag of Words (bolsa de palabras) deriva del hecho de que el orden de las palabras en la historia no influye en la proyección. Es considerablemente más rápido que la otra arquitectura y funciona bien sobre las palabras más frecuentes. Sin embargo el autor menciona que Skip-Gram retorna mejores resultados para representar las relaciones semánticas.

Existen otras técnicas bien conocidas además de word2vec, con las que se pueden obtener buenos resultados. De ellas las que queremos mencionar en

este contexto son GloVe [20] y Fast Text del laboratorio de investigación de Inteligencia Artificial de Facebook (FAIR) [21].

GloVe

GloVe (abreviación de Global Vectors) es un proyecto de código abierto creado por Standford, se diferencia de word2vec por ser un modelo estadístico basado en conteos (en vez de ser uno predictivo). GloVe aprende mediante la construccion de una matriz de co-ocurrencia (palabras X contexto), donde básicamente cuenta las ocurrencias de una palabra en el contexto dado. Dado que dicha matriz presenta un gran tamaño se le aplica una función de costo minimizada asociada a los vectores y a la matriz para obtener representaciones vectoriales de baja dimensionalidad. El principal beneficio que nos brinda es que es posible paralelizar la implementación, por lo que es más sencillo trabajar sobre volúmenes de datos más grandes, además de que logra mejores resultados mucho más rápido.

FastText

Por otro lado, Fast Text de código abierto, también esta implementado como una red neuronal para obtener representaciones vectoriales de las palabras y para clasificación de textos. Facebook pone a disposición de los usuarios modelos pre-entrenados para 294 idiomas diferentes usando Wikipedia como corpus. Combina varias herramientas del PLN, tales como bag of words, bag of n-grams, intercambio de información entre clases a través de una representación oculta, softmax jerárquico para aumentar la velocidad de los cálculos, etc. En líneas generales, es parecido a Skip-Gram, pero en lugar de solamente quedarse con un producto puntual entre dos vectores utiliza además lo de los caracteres correspondientes de n-gramas de 3 a 6. Por lo tanto, el vector de una palabra será la colección de la palabra junto con esos caracteres. Para ilustrar con un ejemplo, el siguiente vector correspondería a la palabra "water".

```
water = { wat, ate, ter, wate, ater, water }
```

A partir de esto podemos decir que es como la extensión del trabajo de Skip-Gram. Gracias a su simplicidad el entrenamiento es rápido y no requiere de preprocesamiento o de supervisión.

2.2.6. Modelo de lenguaje

Dada la expresión "***El cuaderno está en la ...***" y dadas las siguientes oraciones que la completan:

1. "***El cuaderno está en la mochila.***"
2. " ***El cuaderno está en la perro.***"

Es más frecuente encontrar la primera que la segunda. El objetivo de un modelo de lenguaje es modelar, valga la redundancia, este fenómeno, asignándole una probabilidad a cada oración, según qué tan probable es que aparezca en un texto del lenguaje.

En general, para construir un modelo de lenguaje se utiliza un corpus de gran tamaño, para ver qué combinaciones de palabras aparecen frecuentemente allí.

Contar directamente la cantidad de frases o segmentos que aparecen dentro del corpus en general no es una buena idea, ya que puede haber algunas frases que no aparezcan en el corpus pero que igual sean comunes. Este problema se puede resolver de varias maneras, una de ellas podría ser aplicando técnicas de suavizado, donde por ejemplo siempre se le suma una fracción no nula a todas las probabilidades.

Trabajando sobre un corpus grande, contar las ocurrencias de cierta frase es una tarea lenta y tediosa. Además de que es raro ver una frase apareciendo exactamente igual en muchos textos (aunque los significados de las frases sean los mismos). Por estas razones, se recurren a otros métodos distintos de los manuales, como bigramas, trigramas, cuatrigramas, etc. En nuestro proyecto utilizamos el de trigramas.

Modelo de trigramas

Dada una oración O con las palabras $W_1, ..., W_n$. El modelo de trigramas consiste en asumir que

$$P(O) = \prod_{i=1}^{n+2} P(W_i)|P(W_{i-2} - W_{i-1})$$

Donde $W_{-1} = W_0 = W_{n+1} = W_{n+2} = \$$, un carácter especial para marcar el fin y comienzo de una oración y $P(A|B - C)$ es la probabilidad de que la siguiente palabra en un texto sea A, asumiendo que las anteriores fueron B y C.
El modelo de n-gramas tiene dos supuestos. El primero es que las ocurrencias de las palabras son independientes de lo que pasó antes. El segundo es el hecho de que los sucesos son independientes (la probabilidad de una oración es el producto de n-gramas).

Lo que queda para construir el modelo de lenguaje es estimar las probabilidades $P(W|W' - W'')$. Una forma de hacer esto es utilizar el estimador

de máxima verosimilitud (o método de frecuencias relativas). Este método consiste en tomar un corpus, contar las ocurrencias de los trigramas que aparecen sobre el total de trigramas que empiezan con $W' - W''$, y normalizar.

$$P(W|W' - W'') = \frac{C(W' - W'' - W)}{\sum_{W*} C(W' - W'' - W*)}$$

Aplicaciones

Entre las diferentes aplicaciones que tienen los modelos de lenguaje se encuentran:

- Predicción de palabras, probabilidad de una oración
- Traducción automática
- Reconocimiento de escritura manuscrita
- Corrección de faltas de ortografía (comúnmente es la más utilizada)

En particular, en nuestro proyecto utilizamos trigramas para poder encontrar la secuencia con mejor probabilidad entre varias secuencias de palabras donde las mismas solo difieren en una palabra.

2.3. Trabajos relacionados

"Natural Language Processing for Enhancing Teaching and Learning" de Diane Litman [22] es un artículo relevante por explicitar el rol del PLN en la enseñanza de idiomas. Además, el artículo nos fue clave para orientarnos en las oportunidades y desafíos de esta disciplina. En él se proponen tres roles principales para los que el PLN puede ser utilizado dentro de la investigación educativa: aprender aspectos del lenguaje en sí mismo, aprender sobre cualquier área de estudio de interés y satisfacer cualquier necesidad dentro de la enseñanza de lenguas. Dentro de esos tres roles nosotros nos centraremos en el primero y en el último, que son los que conforman nuestro proyecto.

A continuación exponemos algunas de las ideas de este artículo y distintos sistemas que utilizan el PLN para la enseñanza.

2.3.1. Natural Language Processing for Enhancing Teaching and Learning - Diane Litman

El interés en el procesamiento de lenguaje natural para el uso de aplicaciones dentro de la enseñanza ha ido creciendo en los últimos años, ya que se ha comprobado que puede mejorar las técnicas existentes que se utilizan en esta área. Se proponen tres principales roles de uso para el PLN y nuestro proyecto se basa principalmente en el primero y el tercero:

1. Enseñar y aprender temas relacionados con el lenguaje en sí mismo, como puede ser la escritura, comprensión lectora, etc. El procesamiento de lenguaje natural puede usarse aquí, por ejemplo, para puntear textos de estudiantes tomando en cuenta reglas de la gramática.
2. Usar el procesamiento de lenguaje para enseñar y aprender cualquier materia o área de estudio de interés. Por ejemplo a través del uso del PLN, las tutorías o clases particulares se pueden diseñar de forma rentable y escalable con tecnologías de diálogo.
3. Satisfacer las diferentes necesidades del aprendizaje de lenguas por parte de estudiantes, profesores, investigadores, etc. utilizando el procesamiento de lenguaje natural. Como algunos ejemplos encontramos el procesamiento de texto en la web para personalizar materiales, generar ejercicios, preguntas o automatizar un sistema completo de tutorías.

La autora expone que cuando se utiliza el procesamiento de lenguaje natural en la educación nos encontramos con verdaderos desafíos. En primer lugar está el problema de adecuación de textos para los usuarios. Muchas de las herramientas existentes fueron entrenadas con textos bien formulados (por ejemplo textos periodísticos) por lo que no se desempeñan bien con textos escritos por estudiantes que seguramente contengan errores.

En segundo lugar tenemos el reto de identificar las características significativas de los textos a la hora de calificarlos, esto es, cuando se va a puntuar es necesario encontrar aspectos que sean relevantes y que hay que tener en cuenta a la hora de la evaluación. A modo de ejemplo, podría ser útil realizar un recuento de palabras mientras se elabora un ensayo pero no sirve para posteriormente calificar aspectos gramaticales u ortográficos.

Por último, no hay que olvidarse de contar con soluciones técnicas adecuadas para sistemas interactivos, debido a que los algoritmos de procesamiento de lenguaje natural están integrados a este tipo de aplicaciones es necesario contar con respuestas en tiempo real y escalabilidad de la solución, por lo que su implementación puede ser más compleja.

Respecto al rol número tres, el artículo sugiere que de esta forma se logra aprovechar mayor cantidad de textos. Para este tipo de programas, se pueden

utilizar grandes volúmenes de textos que se encuentran en la web, además de discursos de voz y otros contenidos multimedia. La principal ventaja que se obtiene al automatizar tareas como la creación de actividades o materiales de evaluación es que tanto el esfuerzo humano como el tiempo invertido se ven considerablemente disminuidos. Además, estos materiales luego se pueden adaptar a las diferentes necesidades o aptitudes que tengan los estudiantes en el aula.

Algo interesante que la autora comenta es que se está estudiando la simplificación de textos como un método para poder reutilizar los materiales existentes en todos los niveles de estudiantes [23]. Con respecto a la evaluación se están explorando métodos basados en el procesamiento de lenguaje natural para generar automáticamente preguntas de múltiple opción, banco de palabras (o ejercicios para completar) y otros tipos de preguntas en el dominio de los distintos temas de estudio (la mayoría de estos estudios se encuentran en varios de los trabajos mencionados más adelante en la sección 2.3.4).

Para culminar con el artículo la autora menciona que evaluar un programa de procesamiento de lenguaje natural sobre la educación en los salones de clase hoy en día es muy poco común pero, a medida que se continúe investigando y desarrollando, será cada vez más habitual. Esto permitirá determinar, por ejemplo, la tolerancia de las tecnologías a los errores, evaluar si los métodos pueden ser generalizados entre los profesores, estudiantes, escuelas, etc., y sobre todo permitirá aumentar las colaboraciones entre investigadores para continuar con el desarrollo de estas herramientas.

2.3.2. Language Muse

Language Muse [24] es una aplicación web que fue diseñada con el objetivo de brindar apoyo profesional a docentes a la hora de preparar las lecciones, haciendo énfasis en los elementos del idioma inglés y considerando el hecho de que haya alumnos que puedan tener poca o nula familiaridad con el inglés. Pretende ayudar a los estudiantes de inglés en el desarrollo de una profunda comprensión lectora en las diferentes áreas que componen la currícula estudiantil, tales como matemática, ciencias, sociales y arte. Todo esto se trata de lograr a través de la creación automática de ejercicios a partir de textos que pueden ser editados por los docentes. Utiliza métodos y recursos del procesamiento de lenguaje natural para generar los ejercicios. En la figura 2.3 se observa una imagen de la interfaz gráfica al momento de generar un ejercicio.

Luego de su lanzamiento se realizaron pruebas de usuario sobre su versión piloto con profesores reales. Las mediciones indicaron que los maestros aumentaron en gran medida su conocimiento lingüístico sobre vocabulario y

relaciones de oraciones y de discurso [25]. Actualmente se encuentra integrado con tres programas orientados al desarrollo profesional de los docentes en Estados Unidos.

Lo que diferencia a Language Muse de otras aplicaciones que utilizan PLN es que está orientada en parte al desarrollo profesional docente de forma que comprendan las características lingüísticas de los textos, en vez de estar dirigida solamente a los alumnos.

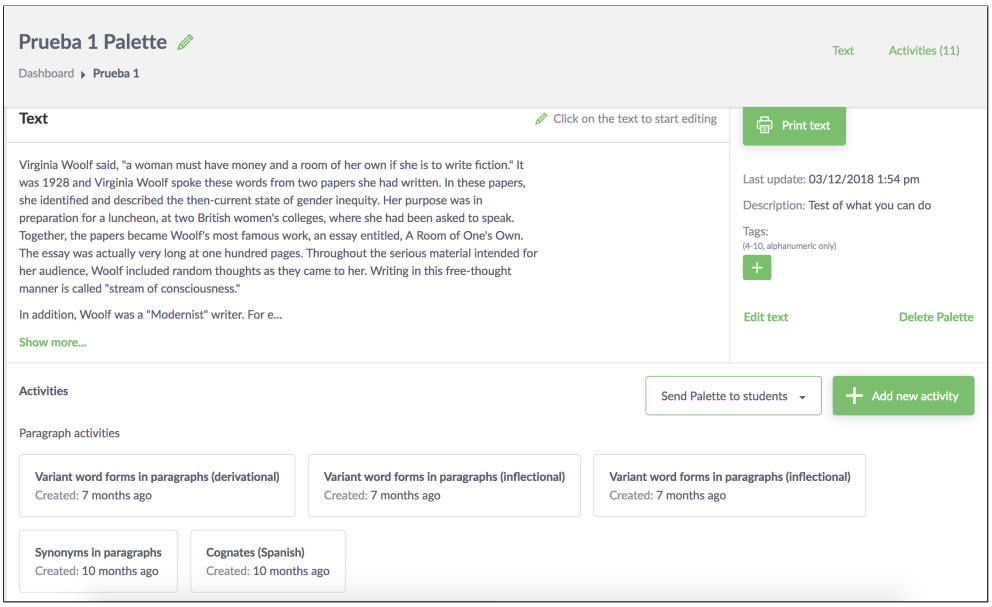

FIGURA 2.3: Interfaz gráfica de Language Muse al seleccionar un texto para crear una actividad

2.3.3. Sistemas CALL

El aprendizaje de idiomas asistido por computadora (también conocido como e-learning), de ahora en adelante CALL por su sigla en inglés, refiere al estudio y a la puesta en práctica del rol que desempeñan las tecnologías de la información y de las comunicaciones para la enseñanza de lenguas (como su nombre lo indica). Recurre a varios tipos de actividades, cada una con diferentes materiales, como por ejemplo tutoriales y cursos en línea, juegos educativos, procesadores de texto, reproductores multimedia, navegadores de internet, entre otros [26].

Dentro de los sistemas CALL, encontramos los que usan procesamiento de lenguaje natural para brindar una mejor presentación de la información lingüística y sobre todo brindar retroalimentación automática. Estos sistemas se denominan Intelligent Computer-Assisted Language Learning (ICALL). Recientemente los últimos avances se han concentrado en esta área. A su vez dentro de ICALL tenemos los sistemas ATICALL que suman la característica de trabajar solamente sobre textos nativos del lenguaje ("Authentic Text ICALL") [27].

En la actualidad la mayor parte de los analizadores construidos se limitan simplemente a los aspectos sintácticos y gramaticales, encargándose por ejemplo de detectar errores en los ejercicios de lectura permitiendo que los profesores dediquen más tiempo a interactuar con los alumnos. Con estos sistemas los estudiantes cuentan con la factibilidad de poder crear oraciones nuevas en el idioma que están interesados, ingresarlas al computador y recibir respuestas como retroalimentación. Esto es lo que diferencia un sistema ICALL de uno CALL.

La importancia de los parsers en los sistemas CALL fue muy discutida en la última década por varios investigadores (Nagata, Matthews, Holland, Maisano, entre otros). Muchos se refieren a las posibilidades y limitaciones que tienen los tutores de lengua basados en parsers. Lo hacen realizando una comparación entre un sistema CALL basado en parsers y otro que no, y concluyen que en el primero los alumnos pueden escribir una gran variedad de oraciones y cuentan con más libertad para desarrollar habilidades para el mejoramiento de la escritura [12].

Hoy en día con lo que se está intentando trabajar más es con la componente semántica de los textos, ya que por el momento estos sistemas no son demasiado eficientes al tratar de averiguar el significado a nivel textual.

A continuación se describen algunos de los más conocidos o populares sistemas ATICALL que se pueden encontrar en la actualidad:

1. WERTi (Working with Real Texts in English in an Interactive Way) [26] es una herramienta que aplica métodos de procesamiento de lenguaje natural para identificar aquellas formas lingüísticas que pueden ser poco conocidas para los estudiantes. Luego de reconocidas, ellos son capaces de interactuar directamente con esas salidas. De esta manera, la herramienta permite realizar mejoras en textos en español, inglés y alemán. Las formas se identifican utilizando códigos de colores, y también se pueden modificar según las necesidades del usuario, para por ejemplo, hacer el texto más complejo. También es posible generar algunas actividades dentro del texto, como puede ser elegir la opción correcta para una palabra o rellenar un espacio en blanco de una oración. Parte de la interfaz gráfica se puede apreciar en la figura 2.4.

2. SmartReader [26] es un asistente de lectura que resalta contenido que le puede interesar al estudiante a partir de un analizador de textos, ofreciendo así definiciones de las palabras e información enciclopédica sobre entidades. En base a esta información, realiza preguntas sencillas sobre las entidades (siempre que las respuestas se encuentren explícitamente dentro del contexto). Además informa al usuario sobre los elementos de las oraciones tales como el sujeto o el predicado.

3. SMILLE (Smart and Intelligent Language Learning Environment) [28] es un sistema ATICALL que mejora textos que se encuentran en páginas

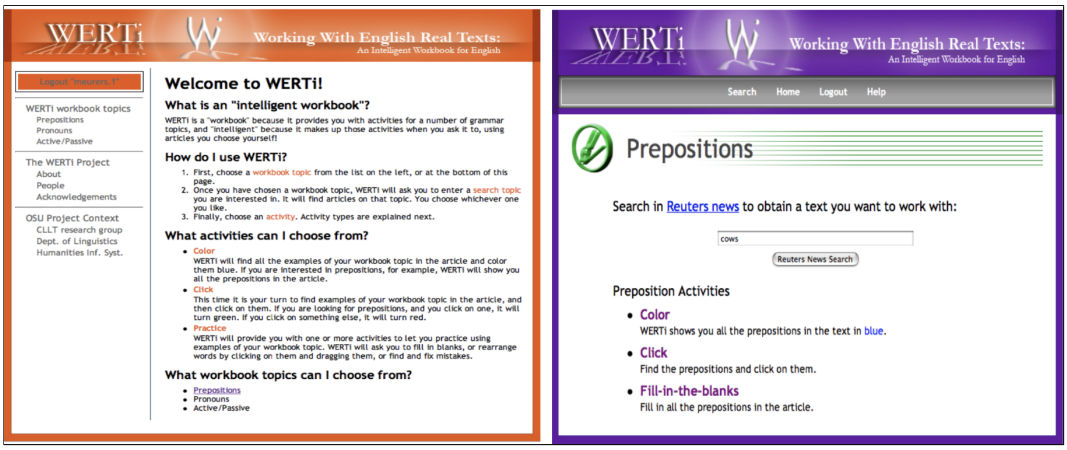

FIGURA 2.4: Interfaz gráfica de WERTi

y sitios web. El mismo recibe un documento como entrada y se encarga de realizarle un procesamiento combinando un analizador y reglas de detección de estructuras gramaticales para mejorarlo de acuerdo a las necesidades específicas de aprendizaje que tenga el usuario. SMILLE ayuda de esta forma a la comprensión lectora y proporciona información sobre los significados de las palabras en el texto mientras que, al mismo tiempo, da conciencia sobre las estructuras gramaticales. En el caso de que el alumno se encuentre estudiando verbos conjugados en pasado simple en el texto le aparecerán resaltados todos esos elementos para llamar su atención. De momento, sólo está implementado para el inglés. Uno de sus principales objetivos es que los estudiantes no encuentren limitaciones a la hora de elegir un texto de forma de mantenerlos motivados a lo largo de todo su aprendizaje.
El formato en que SMILLE muestra la información es bastante similar al de los dos sistemas anteriores, pero el alcance y la variedad de las estructuras gramaticales que abarca es significativamente mayor.
Para describir su funcionamiento en pocas palabras, extrae el contenido del texto a partir del HTML de la web, lo analiza usando diferentes analizadores sintácticos (como el parser de Stanford) para agregar etiquetas léxicas y con diferentes reglas logra crear nuevas etiquetas para las estructuras gramaticales. Finalizado esto, una nueva página web es elaborada con la misma información de la página anterior pero usando diferente código HTML y agregándole scripts que permiten hacer modificaciones del texto en tiempo real.
Algo en lo que se está trabajando es en agregar una base de datos para guardar la información buscada y la actividad de los usuarios, ya que podría mejorar notoriamente la experiencia de los mismos y así hacerla más personalizada.

Para finalizar con los sistemas ATICALL, podemos decir que es posible encontrar muchos más, y no solo del tipo que se centre en el resaltado de las estructuras gramaticales. Por ejemplo encontramos el denominado REAP system [29], que también recupera información pero enfocándose en el perfil

del usuario y su nivel de vocabulario. También hay otros sistemas que se encargan de la mejora de la legibilidad o la simplificación de los textos.

2.3.4. Otros trabajos

Para terminar esta sección mencionamos ideas de otros trabajos que contribuyeron a la hora de idear y desarrollar la aplicación.

Hay varios artículos que son relevantes para la tarea que nos compete. Muchos están basados en generación automática de preguntas (los creadores de Language Muse se basaron en estos artículos). En [30] se discute la generación de tests múltiple opción, mientras que en [31] explica cómo generar preguntas de comprensión del vocabulario utilizado en cierto texto. Otro ejemplo se encuentra en [32], que explica un método para generar muchas preguntas sobre un texto y luego elegir las mejores.

En [33] se investigan los desafíos que implica modificar los modelos clásicos de evaluación.

También hay investigaciones que buscan evaluar de forma automática la coherencia del discurso, como en [34] que se centra en brindar retroalimentación a respuestas escritas.

Por último, también encontramos artículos sobre resúmenes de textos escritos en [35].

Si bien todos estos temas están relacionados a nuestro proyecto, no fueron centrales para su desarrollo.

3. Requisitos y Esquema de la solución

3.1. Requisitos relevados

Los requisitos surgieron de las funcionalidades y limitantes planteadas al comienzo del proyecto por los interesados, junto con los que fuimos relevando durante el desarrollo.

En nuestro proyecto fue necesario construir dos aplicaciones (una orientada a los docentes y otra a los estudiantes), por lo tanto, algunos requisitos son específicos de una aplicación en particular.

3.1.1. Requisitos funcionales

Generación de ejercicios a partir de texto ingresado

El primer y fundamental requisito fue que los docentes puedan generar ejercicios a partir de un texto ingresado en inglés, para luego distribuirlo con los estudiantes. Decidimos hacer cuatro tipos de actividades, basándonos en los estándares clásicos de ejercicios de inglés acorde a los niveles establecidos, como por ejemplo llenar los espacios en blanco (fill in the blanks en inglés), unir opciones correctas, etc.

La aplicación de los estudiantes debe de contar con la opción de corregir automáticamente cada ejercicio, para que el estudiante pueda realizarlos sin necesidad de la presencia del docente.

Colección de textos

Contar con una colección de textos fue una necesidad que surgió por parte de los docentes, ya que a veces resulta difícil encontrar textos adecuados para la generación de ejercicios. Además, se espera que los textos ingresados por los docentes sean guardados en una base de datos, de forma de poder compartir los textos entre todos los docentes.

Edición de ejercicios

Al ser una herramienta construida en base a métodos del PLN, los cuales no son totalmente precisos, era necesario que fuera supervisada por un docente. Para lograrlo era imprescindible entonces contar con la posibilidad de editar los ejercicios una vez que fueran creados (previamente a que se guarden). Esto le permite al docente corregir errores que presenten los ejercicios o mejorarlos en caso de que alguna opción no sea acorde al nivel de inglés de sus alumnos. Por lo tanto, los maestros deberían ser capaces de poder cambiar o eliminar opciones dentro de un ejercicio.

Ver y editar diccionario

La plataforma cuenta con su propio diccionario, que es utilizado en algunos de los ejercicios. Nosotros generamos nuestro propio tesauro infantil (explicado más adelante en sección de recursos generados 4.1.1), por lo que puede contener errores o pueden faltar definiciones. El poder contar con el listado y la edición de las definiciones nos pareció de gran utilidad.

3.1.2. Requisitos no funcionales

Sin conexión a internet

Desde el comienzo del proyecto nos plantearon el requisito de que las aplicaciones deberían funcionar sin conexión a internet dada la realidad de las escuelas rurales. En un principio, nuestro objetivo era poder tener ambas aplicaciones funcionando sin conexión a internet. Pero, durante el transcurso del proyecto, nos encontramos con que la aplicación del docente inevitablemente demandaría una gran capacidad de procesamiento y espacio de almacenamiento, lo cual generaba dificultad para la distribución de la misma.

A pesar de que logramos disminuir significativamente el tamaño y la capacidad de procesamiento, sabíamos que tener un servidor donde la aplicación pueda acceder con internet nos traería muchos beneficios como compartir los recursos generados entre los docentes, mejorar el diccionario sin necesidad de actualizar la aplicación, etc. Al ver la realidad de las escuelas rurales nos encontramos con que muchas tenían conexión a internet, aunque a veces limitada o intermitente. Considerando que el maestro puede preparar los ejercicios con antelación y guardarlos en algún dispositivo físico como un pendrive, esto no nos pareció una gran limitante.

Luego de hablar con los interesados del proyecto, nuestra resolución para este requisito fue que los estudiantes puedan utilizar la aplicación sin internet en caso de ser necesario y que la aplicación de docentes requiera de internet

en todo momento para generar los ejercicios, pero permitiendo almacenar los mismos en sus computadoras para trabajar luego en clase sin conexión.

Tiempo de procesamiento

Que el tiempo de procesamiento sea bajo fue un requisito no funcional implícito en la realidad planteada. La única forma de que nuestra herramienta sea práctica es logrando que los tiempos de generación de ejercicios no sean excesivamente grandes. Si bien el tiempo de procesamiento va a depender del texto ingresado y de la capacidad del servidor, en la sección de experimentación (ver 4.3) mostramos métricas con los tiempos arrojados por diferentes textos, consideramos que los tiempos de espera para la generación de ejercicios fueron aceptables.

Compatibilidad con Plan Ceibal

Tanto la aplicación de los docentes como la de los estudiantes debían poder ser ejecutadas en las computadoras o tablets brindadas por el Plan Ceibal [36] a los maestros y alumnos.

Niveles de inglés

Como se explicó previamente en la introducción, la herramienta está enfocada a un rango específico de niveles de inglés (niveles pre-A1 y A1 del Marco Común Europeo de Referencia [8]). Por lo tanto, el resultado final de los ejercicios debe estar dirigido a esos niveles de inglés especificados, es decir, un estudiante que se encuentre en esos niveles debe ser capaz de realizar el ejercicio.

Usabilidad

Como último requisito decidimos que nuestras aplicaciones fueran fáciles de usar, sobre todo orientadas a personas sin mucha experiencia en el uso de computadoras. De esta forma motivaríamos a los docentes a utilizarlo. Ambas aplicaciones cuentan con pocos pasos, ya sea para la generación o la resolución de ejercicios y con una interfaz bastante intuitiva. Este requisito se reforzó durante las reuniones con los maestros, ya que ellos mismos lo confirmaron en repetidas ocasiones, por ejemplo diciendo que les gustaría que la interfaz de la aplicación fuera con un simple fondo en color blanco y con pocos botones.

3.2. Esquema general de la solución

Para nuestra solución implementamos dos aplicaciones con fines distintos, una para el uso del docente y otra para los estudiantes. En la primera, el docente sube un texto en inglés o selecciona uno desde la base de textos donde ya están almacenados. Luego selecciona el tipo de ejercicio que quiere utilizar para trabajar con sus alumnos y el sistema genera automáticamente el ejercicio y lo despliega en pantalla junto con las respuestas. Una vez generado el ejercicio se tiene la posibilidad de guardarlo para que sea visible en la aplicación del estudiante. En la imagen 3.1 se puede apreciar el flujo de trabajo de los maestros para generar un ejercicio con los pocos pasos que tiene.

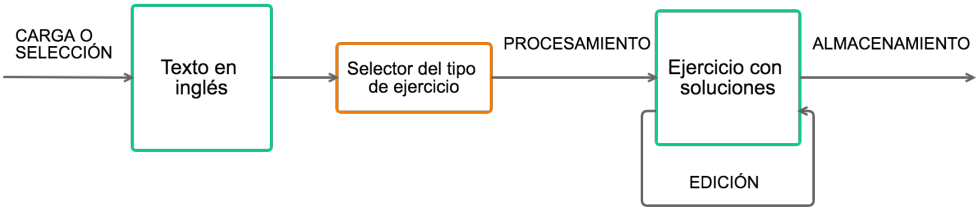

FIGURA 3.1: Flujo para la generación de un ejercicio.

Por otro lado, el alumno ve el nuevo ejercicio que le indica el docente y lo resuelve. Por último corrige el ejercicio y se entera si lo ha resuelto correctamente o ve los errores que ha cometido. Las actividades entonces son generadas por los docentes y los estudiantes por su parte son los encargados de resolverlas, ya sea en la clase o desde su casa.

Ejercicios implementados

A partir de los recursos que construimos y teniendo en cuenta todos los requisitos mencionados logramos implementar cuatro tipos de ejercicios que utilizan diferentes recursos provistos por el PLN y elementos del lenguaje.

Contamos con un ejercicio de conjugación de verbos, en el cual dado un verbo hay que seleccionar el que se encuentra en el tiempo verbal correcto. Considerando los niveles de inglés sobre los que estamos trabajando, optamos por brindar una pequeña ayuda a los estudiantes indicando el tiempo verbal de la palabra a buscar. Para su elaboración es necesario durante el procesamiento del texto primero dividir el texto en tokens (que en nuestro caso se corresponden con las palabras) para luego aplicar POS tagging para identificar todos los verbos. Aquellos que pertenecen al vocabulario de la aplicación se guardan para ser utilizados en el ejercicio. Usando las etiquetas morfológicas del tagging inicial se logra obtener el tiempo verbal en el que se

encuentra cada verbo y se consiguen las posibles conjugaciones, de las cuales se eligen solamente tres, para utilizarlas como los distractores. Un ejemplo de este ejercicio es el siguiente:

"Yesterday, I (1) _____ *(past)* to the supermarket. There I (2) _____ *(past)* some cheese."

1. **went**, gone , going, goes
2. buy, buying, **bought**, buys

Otro ejercicio utiliza sustantivos extraídos de un texto y el reto consiste en unir cada uno con su correspondiente definición dados dos conjuntos, uno de sustantivos y otro de significados. Nuevamente es necesario tokenizar para dividir el texto en palabras para aplicar POS tagging, sólo que esta vez el objetivo es reconocer todos los sustantivos en lugar de verbos. Aquellas palabras que no son identificadas como sustantivos se descartan automáticamente. Con el conjunto obtenido se busca para cada sustantivo su definición dentro de un diccionario generado y si no tiene ninguna porque no se encuentra en el diccionario, se descarta. El recurso diccionario o tesauro infantil (ver 4.1.1) contiene definiciones sencillas de palabras que están orientadas al entendimiento de los niños, fue conseguido y manipulado a través de un diccionario que estaba en la web. En caso de que el sustantivo tenga más de un significado nos enfrentamos a un problema de Word Sense Disambiguation, y se resuelve como se explica más adelante.

Para ejemplificar, a partir de las frases **"The moon hides the stars in the sky. The moon shines in the city."** se creó el ejercicio que contiene las siguientes palabras y sus definiciones:

1. **Sky**: The _____ is the air or space above the earth . During the day, the _____ usually looks blue or gray . At night the _____ looks mostly black.
2. **Moon**: A _____ is a very large object that goes around a planet in a circle . Our earth has only one _____, but some planets have many _____.
3. **City**: A _____ is a place where many people live and work . A _____ is a kind of town, but a _____ is usually a very large town with many buildings and streets.

Un ejercicio al que llamamos de hipónimos o de categorías, consiste en que los alumnos encuentren la categoría adecuada para cada palabra dado un conjunto de palabras y otro de categorías. Por ejemplo si tenemos la palabra **red** entonces la categoría correcta es **colours**. Otra vez se procede a tokenizar el texto para trabajar a nivel de las palabras. Luego se filtran las

palabras para solo tener en cuenta aquellas cuya clase gramatical sea sustantivo o adjetivo. A partir de una lista previamente armada de categorías de concepto (que son básicas dentro del idioma inglés) se tienen que encontrar las palabras que pertenecen a alguna de las clases. Para llevar esto a cabo se utiliza una ontología, que sirvió para reconocer si existe la relación de hiponimia entre cada palabra y las categorías. En caso de que exista se agrega la palabra y la categoría a los conjuntos con los que los alumnos posteriormente tendrán que trabajar. Por ejemplo, para la frase **"He puts on his green shirt and his blue jeans"** se crea el ejercicio:

- **Clothing**: shirt, jeans
- **Colors**: green, blue

Por último contamos con un ejercicio sobre el uso del lenguaje, conocido también como ejercicio de Use of English (uso del inglés en español), que tiene como objetivo completar los espacios en blanco de un texto con la palabra correcta. Para cada espacio en blanco se tienen cuatro opciones, una correcta y tres distractores. Con este ejercicio se pretende que los alumnos aprendan a utilizar el vocabulario dado un contexto y que sepan identificar la palabra correcta para cada oración. Lo más difícil consistió en encontrar los distractores adecuados, ya que al momento de ser sustituidos la oración debe ser semánticamente incorrecta pero a la vez sintácticamente correcta ya que de lo contrario el ejercicio pasaría a ser demasiado fácil de resolver. Al mismo tiempo los distractores no pueden ser muy incoherentes con el contexto. Para lograrlo construimos un modelo de lenguaje y un modelo de word embeddings. A continuación mostramos un ejemplo de este ejercicio:

"My cat (1) _____ pretty and soft. My (2) _____ too."

1. **is**, walks , picks, swims
2. water, carrot, **dog**, guitar

Luego de tokenizar y taggear, otra vez con los mismos recursos, se tienen en cuenta solamente los verbos, sustantivos y preposiciones que se encuentran dentro del texto. Iterando sobre las palabras, se eliminan aquellas que no pertenecen a nuestro vocabulario generado y dado que varias palabras con las categorías gramaticales deseadas pueden aparecer en un mismo enunciado, se conserva una sola palabra, aquella que tenga más frecuencia en el vocabulario según nuestra lista de palabras frecuentes. Para obtener los distractores de cada palabra se aplican una serie de diferentes filtros sobre todas las palabras de nuestro vocabulario. En el caso de que la palabra sea un verbo, los distractores que se retornan están conjugados en el mismo tiempo verbal.

Edición de los ejercicios

Es imposible saber lo que el docente quiere exactamente para la actividad que planea. Por eso permitimos que todos los ejercicios se puedan modificar antes de guardarlos. Otra razón es que la herramienta no es 100 % efectiva y puede fallar, por lo que se necesita que una persona revise los ejercicios y los edite en el caso de que sea necesario. La mayoría de los aspectos del ejercicio son editables y con esto también se permite el borrado de opciones. Además, todos los docentes tienen acceso a los recursos generados de vocabulario, diccionario y textos, donde los dos primeros también son modificables.

Requisito de conectividad

Sobre el requisito de conectividad, en el caso de que los estudiantes no tengan internet se brinda la posibilidad de exportar los ejercicios al guardarlos. Al exportar, se genera un archivo que el docente entrega al alumno (por medio de internet o un pendrive por ejemplo). El alumno, desde su aplicación, importa el archivo y visualiza el ejercicio. La aplicación corrige el ejercicio en tiempo real, sin necesidad de internet ya que las soluciones se guardaron al momento de generar el ejercicio.

4. Solución desarrollada

En esta sección hacemos una explicación exhaustiva sobre el trabajo realizado. Esto incluye los recursos generados y los ejercicios en sí mismos. Explicamos los diferentes aspectos de la plataforma y cuáles fueron las principales dificultades y cambios realizados durante el proyecto.

4.1. Recursos generados

Junto con la utilización de las técnicas y herramientas previamente detalladas en el marco teórico, la elaboración de todos los recursos fue una tarea fundamental para poder generar los ejercicios.

La complejidad de algunos ejercicios que se presentan más adelante nos llevó a iterar varias veces sobre los recursos para poder definir una adecuación correcta de los mismos en el contexto de la enseñanza de un idioma. Por ejemplo, si sólo hubiéramos entrenado nuestro modelo de lenguaje con un corpus periodístico entonces habríamos obtenido palabras demasiado complejas o formales para los estudiantes.

Además, los recursos generados están sujetos a un rango de niveles para el idioma inglés especificados por ANEP, los cuales son pre-A1 (Starters) y A1 (Movers) según el Marco Común Europeo de Referencia para las lenguas [8].

4.1.1. Tesauro infantil

Un requisito para los ejercicios era tener un tesauro infantil en inglés que contenga la definición y la traducción de la palabra al español. La dificultad para encontrarlo fue que tenía que ser infantil o con palabras simples pero a la vez lo suficientemente completo como para encontrar palabras a partir de un texto ingresado por el docente.

Por otro lado, las definiciones debían ser comprensibles por alguien que está estudiando el lenguaje y tampoco se podían usar otras palabras complejas para definir la palabra. Es decir, que no podía ser un diccionario convencional.

Wordsmyth [37] es un diccionario y tesauro online enfocado a la educación. Entre todos los recursos que ofrece tiene un tesauro infantil [38]. Al ser de código abierto provee una llamada a un servidor para obtener un archivo en formato JSON con todas las palabras y sus definiciones. Creamos un script para poder recorrer el archivo JSON y guardar todas las definiciones en nuestra base de datos. En total guardamos **3567 definiciones.**

Como mencionamos, este es un tesauro enfocado en el aprendizaje por lo tanto muchas de sus definiciones pueden contener la propia palabra, esto es intencional para que el estudiante pueda comprender cómo se usa la palabra en un contexto.

Comparando la palabra **"attach"** de nuestro tesauro contra un diccionario convencional como el diccionario de Cambridge [39] tenemos el siguiente resultado:

Definición Tesauro Wordsmyth

> *When you attach two things, you join them or stick them together.*

Definición Cambridge

> *To fasten, join, or connect something*

El tesauro de Wordsmyth, además de tener una definición más simple nos brinda ejemplos, traducciones al español y posibles categorías gramaticales de la palabra.

4.1.2. Lista de palabras Starters y Movers

A partir de los materiales que nos brindaron los funcionarios de ANEP, construimos a mano una lista de palabras indicadas para los niveles Sarters y Movers de la Universidad Cambridge. Esta lista es simplemente un archivo en formato JSON y es utilizada en la mayoría de nuestros ejercicios ya que contiene el mínimo vocabulario que se espera que los alumnos conozcan. La lista de palabras la agrupamos por su categoría gramatical (indicado en los materiales brindados) para que luego sea más fácil su procesamiento. Las categorías utilizadas fueron verbos, determinantes, preposiciones, adverbios, conjunciones, adjetivos, pronombres y sustantivos.

4.1.3. Palabras frecuentes

Para complementar el vocabulario de Starters y Movers, utilizamos una lista de palabras frecuentes. Esto nos da un vocabulario base para que la plataforma utilice, independiente al diccionario de la base de datos.

La lista de palabras frecuentes la tomamos de Wiktionary [40], un proyecto colaborativo que desarrolla diccionarios para diferentes idiomas. Está basada en los libros de Project Gutenberg [41] y se creó en 2006.

4.1.4. Vocabulario

Era necesario contar con un conjunto de palabras acotado que determine el vocabulario para los ejercicios. Esta lista de palabras fue utilizada para la generación del modelo de lenguaje y en cada ejercicio es un filtro de las palabras ingresadas a través del texto.

El vocabulario fue formado por la unión de la lista de palabras para los niveles Starters y Movers que nos brindó ANEP junto con las conjugaciones de todos los verbos en esa lista y las cinco mil palabras más frecuentes de la lista de palabras frecuentes. Utilizamos un script para unir las palabras y la librería Pattern [42] para conjugar los verbos en tercera persona, pasado y pasado perfecto.

Debido a que el vocabulario es utilizado cada vez que se genera un nuevo ejercicio es un recurso dinámico, es decir, que puede ser modificado en cualquier momento.

4.1.5. Corpus

Uno de los grandes problemas con los que nos encontramos en este proyecto fue la escasez de corpus para los recursos que generamos. Además, la mayoría de los textos y ejercicios que los maestros utilizan están en formato PDF, lo que dificulta la recolección de documentos y la lectura por la baja precisión de las herramientas de reconocimiento óptico de caracteres (OCR).

Al principio buscamos un corpus con lenguaje simple como uno de cuentos para niños o Simple Wikipedia [43]. Sin embargo, la cantidad de datos que pudimos conseguir no eran suficientes para los recursos que queríamos generar.

Simple Wikipedia es una versión de Wikipedia reducida con el objetivo de facilitar la lectura a niños o personas que estén aprendiendo un idioma. Utilizamos este corpus en etapas de desarrollo de la plataforma ya que su tamaño

es pequeño en comparación con el corpus completo de Wikipedia. Simple Wikipedia pesa 10MB comprimido contra Wikipedia con 4.26GB comprimido.

Para obtener las palabras más frecuentes utilizamos el Project Gutenberg [41]. Project Gutenberg contiene una gran variedad de libros en texto plano. Sin embargo, son libros viejos cuyo copyright expiró (sobre todo libros de antes de 1923), por lo que contienen lenguaje antiguo, que muchas veces no refleja el idioma de hoy en día. Por otro lado, a diferencia de Wikipedia, contiene muchas palabras adecuadas para niños y para el aprendizaje de un idioma, por la cantidad de libros de niños que se encuentran allí.

Tanto el modelo de lenguaje como el modelo de word embeddings necesitan corpus grandes, por lo que al final optamos por utilizar el corpus completo de Wikipedia para generar los mismos.

Wikipedia contiene una gran variedad de vocabulario. Pero muchas veces es muy técnico e incluye palabras complicadas o en otros idiomas. Para solucionar esto, en lugar de modificar los corpus, decidimos utilizar el corpus completo pero aplicando diferentes estrategias de reducción y selección al generar los modelos de lenguaje y word embeddings. Estas estrategias son detalladas más adelante en las secciones de modelo de lenguaje y word embeddings.

Los corpus de Wikipedia y Project Gutenberg nos dieron equilibrio para generar modelos precisos del idioma inglés y obtener palabras adecuadas para niños y/o estudiantes del idioma.

Construcción del corpus

Para construir el corpus de Wikipedia utilizamos el módulo **WikiCorpus** de la librería Gensim [44]. Este módulo genera el corpus en texto plano (formato .txt) utilizando un dump de la base de datos de Wikipedia en formato xml. Wikipedia tiene disponibles los dumps actualizados para descargar desde Wikimedia [45].

En primer lugar descargamos el último dump disponible desde Wikimedia, el mismo pesa aproximadamente 15GB descomprimido. Luego, implementamos un script que toma como entrada el dump descargado para leerlo con WikiCorpus y utilizando el método **get_texts** de WikiCorpus imprimimos la salida en un archivo con extensión txt.

El procesamiento nos demoró 20 horas aproximadamente y nos dio como resultado el corpus en texto plano con un peso de 15GB, pero luego de comprimirlo, tan sólo 4.5GB.

En el Anexo A de procesamiento del corpus se puede ver más detalles del script utilizado para generarlo.

4.1.6. Modelo de lenguaje

Por las características de nuestra plataforma, buscamos un modelo de lenguaje de trigramas para que la oración en general influyera más en la similitud de las palabras. Un modelo de bigramas no nos pareció adecuado ya que sólo tendría en cuenta las palabras contiguas a la hora de decidir la probabilidad. En cambio, cuatrigramas, nos pareció demasiado para las oraciones (que son típicamente cortas) del nivel indicado en los requisitos de la plataforma. Si el n-grama es muy largo es posible obtener más probabilidades nulas (más ceros), lo que se conoce como dispersión, ya que las frases no aparecen en el corpus y por lo tanto directamente son descartadas. Es por estos motivos que utilizamos trigramas.

Nos costó mucho encontrar una librería que pudiera implementar trigramas; por ejemplo, NLTK tiene una librería para n-gramas pero actualmente está deprecada. Finalmente, optamos por una librería llamada KenLM [46] para Python 2. Luego de entrenar el modelo, esta librería tiene un método para obtener la probabilidad de una cadena de palabras (probabilidad logarítmica en base 10).

Al construir el modelo de lenguaje nos encontramos con limitantes de tamaño, consumo de memoria (performance) y funcionamiento del mismo.

KenLM utiliza la salida de un script para calcular los trigramas. Como primera opción hicimos un script simple que procesaba el corpus con NLTK [14] para separarlo en oraciones y lo devolvía como salida para entrenar el modelo.

En la etapa de desarrollo entrenamos el modelo de trigramas utilizando el corpus de Wikipedia Simple [43]. El modelo generado era aceptable para buscar, entre varias oraciones, las dos más distantes en probabilidad (aplicación necesaria para generar los ejercicios), pero no se comportaba muy bien cuando las oraciones eran similares.

Dado el ejemplo de la figura 4.1 se puede ver como la oración ***He eats his dog*** tiene una mejor probabilidad que ***He eats his breakfast***.

Para solucionar esto decidimos entrenar nuestro modelo con un corpus más grande, optando por el de Wikipedia completo. Obtuvimos muy buenos resultados, a continuación, en la figura 4.2 se puede ver el ejemplo anterior analizado con el modelo de lenguaje del corpus de Wikipedia. Todas las probabilidades tienen sentido con la frecuencia que uno utilizaría estas oraciones.

El modelo de lenguaje generado ocupaba aproximadamente 22GB y además consumía demasiada memoria al cargarlo. Por lo que era necesario un servidor muy potente para correrlo y que calcule rápido las probabilidades.

```
>>> score_texto("He eats his breakfast")
-15.595216751098633
>>> score_texto("He eats his dog")
-14.35260009765625
>>> score_texto("He eats his bed")
-15.132365226745605
>>> score_texto("He eats his food")
-15.012808799743652
>>> score_texto("He eats his meal")
-16.082149505615234
```

FIGURA 4.1: Probabilidades calculadas de modelo entrenado con corpus de Wikipedia simple

```
>>> score_texto("He eats his breakfast")
-14.039145469665527
>>> score_texto("He eats his dog")
-15.415968894958496
>>> score_texto("He eats his bed")
-17.2334041595459
>>> score_texto("He eats his guitar")
-16.031147003173828
>>> score_texto("He eats his food")
-14.153310775756836
>>> score_texto("He eats his meal")
-14.21485424041748
```

FIGURA 4.2: Probabilidades calculadas de modelo entrenado con corpus de Wikipedia

Debido a la falta de recursos informáticos, nos vimos forzados a reducir nuestro modelo de lenguaje para que ocupe menos espacio y memoria.

KenLM provee una herramienta para filtrar los n-gramas luego de haber sido generados. El filtro más adecuado para nuestro caso fue filtrar por vocabulario, es decir, si el trigrama no pertenece al vocabulario entonces se descarta.

El vocabulario utilizado fue el generado por nosotros, que contiene las palabras frecuentes, la lista de palabras movers y las conjugaciones de los verbos (ver sección 4.1.4). Una vez construido el vocabulario, aplicamos el filtro de KenLM sobre nuestro modelo. Nuestro modelo reducido cambió de **22GB** a tan sólo **2.4GB**.

Calculando las mismas oraciones de los ejemplos anteriores en el modelo reducido (ver figura 4.3) obtuvimos que las probabilidades son exactamente las mismas que en el modelo no reducido (ver figura 4.2). Esto ocurre porque todas las palabras de la oración pertenecen al vocabulario entonces ningún trigrama que influye en la probabilidad fue eliminado. Esta decisión no afecta negativamente en nuestra plataforma porque las palabras que consideramos en nuestros ejercicios son solamente las que están en el vocabulario.

```
>>> score_texto("He eats his breakfast")
-14.039145469665527
>>> score_texto("He eats his dog")
-15.415968894958496
>>> score_texto("He eats his bed")
-17.2334041595459
>>> score_texto("He eats his food")
-14.153310775756836
>>> score_texto("He eats his meal")
-14.21485424041748
```

FIGURA 4.3: Probabilidades calculadas de modelo reducido con corpus de Wikipedia

Otra observación fue que el corpus generado de Wikipedia con Gensim elimina todos los signos de puntuación. Eso significa que al calcular la probabilidad, si la oración contiene signos de puntación el modelo los va a tomar como caracteres desconocidos y baja la probabilidad. Para resolver esto preferimos remover los signos de puntación de las oraciones cada vez que se calcula la probabilidad. Otra alternativa posible hubiera sido generar nuevamente el corpus pero agregando las puntaciones.

4.1.7. Modelo Word Embeddings

Una de las necesidades de nuestro proyecto era encontrar palabras semánticamente similares a otras. Para realizar esto, inmediatamente pensamos en utilizar word embeddings, donde palabras similares se representan con vectores cercanos.

```
>>> model_wiki.wv.most_similar('bear')
[('bears', 0.5867835283279419), ('lion', 0.44454842805862427), ('rabbit', 0.4440448582172394), ('paw', 0.4400782
585144043), ('ruxpin', 0.4376821517944336), ('dog', 0.43531352281570435), ('wolf', 0.4347929358482361), ('cat',
0.42381277680397034), ('panda', 0.41977792978286743), ('elephant', 0.4193018674850464)]
```

```
>>> model_simple.wv.most_similar('bear')
[('shark', 0.7850852608680725), ('pale', 0.7840403318405151), ('bird', 0.779656171798706), ('whale', 0.761617124
080658), ('deer', 0.7567925453186035), ('purple', 0.7528021335601807), ('cat', 0.7526168823242188), ('eyed', 0.7
505925297737122), ('tailed', 0.7443404197692871), ('crow', 0.7440067529678345)]
```

```
>>> model_achicado.wv.most_similar('bear')
[('bears', 0.5812501907348633), ('dog', 0.5050101280212402), ('lion', 0.49412408471107483), ('rabbit', 0.4889495
074748993), ('paw', 0.4794012904167175 3), ('toothy', 0.47830653190612793), ('monkey', 0.4779062569141388), ('hye
na', 0.4753676950931549), ('wolf', 0.4725094735622406), ('elephant', 0.4703541696071625)]
```

FIGURA 4.4: Los vectores más similares a la palabra "bear", según los tres modelos entrenados.

Para entrenar utilizamos la librería Gensim [47], que es la misma que nos permite utilizar el modelo de vectores.

Manejamos tres conjuntos de vectores distintos, uno utilizando el corpus de Simple Wikipedia y dos con Wikipedia completa. Para evaluar con cuál quedarnos hicimos un balance entre rendimiento y la capacidad de procesamiento que demanda cada uno de los modelos. En la figura 4.4 y el cuadro 4.1 mostramos comparaciones entre ellos.

El de Simple Wikipedia contiene un vocabulario menos abundante y a veces devuelve palabras que no son para nada parecidas. Aunque intuimos que no iba a funcionar bien por la poca cantidad de texto que contiene, decidimos generar embeddings para realizar una prueba de concepto, y porque el entrenamiento es más rápido.
Los vectores de tamaño 400 nos dieron resultados marginalmente mejores que los de 150, pero priorizamos reducir el consumo de memoria de la plataforma, por lo que optamos por utilizar los de 150.

Palabra	**Simple Wikipedia**	**English Wikipedia (Size 400)**	**English Wikipedia (Size 150)**
bear	shark, pale, bird, whale, deer, purple, cat, panda, elephant	bears, lion, rabbit, paw, ruxpin, dog, wolf, cat, panda, elephant	bears, dog, lion, rabbit, paw, toothy, monkey, hyena, wolf, elephant
be	remain, being, handle, damper, happen, surviva, recrystallize, apply, fly, stay	being, have, remain, apply, render, respond, mean, correspond, produce, handle	being, have, been, remain, get, respond, apply, handle, render, produce
ball	bow, batter, hole, stick, puck, rope, button, rocket, finger, flying	balls, puck, bobbled, dribble, pitch, pucks, sliotar	balls, puck, shuttlecock, asketballs, dribble, baseballs, goalpost
pizza	syrup, pudding, ginger, peanut, sausage, soy, tomato	pizzeria, taco, pizzas, burger, sandwiches, hamburgers, pepperoni, calzone, sushi, donut	taco, pizzeria, sushi, hamburgers, nandos, pepperoni, macaroni, burger
quickly	slowly, rapidly, eventually, gradually, slow, fast, weak, easily, soon, hard	eventually, swiftly, soon, slowly, immediately, ultimately, promptly	eventually, swiftly, soon, slowly, rapidly, gradually, finally, ultimately, immediately, easily
most	particularly, more, some, both, only, all, especially, mostly, very, less	more, particularly, highly, fairly, hugely, very, less, especially, many, quite	more, particularly, highly, less, especially, many, very, some, both, only
people	persons, scientists, jews, americans, visitors, skateboarders, muslims, students, workers, everyone	persons, individuals, citizens, villagers, residents, muslims, youths, peoples, jews, christians	persons, individuals, villagers, citizens, youths, residents, muslims, filipinos, peoples, foreigners

CUADRO 4.1: Comparación de los diferentes conjuntos de vectores generados para palabras del vocabulario de Movers.

Parámetros

Gensim provee una serie de parámetros que se pueden ajustar, estos se indican en el cuadro 4.2. El ajuste de parámetros fue un proceso difícil porque la generación de vectores demora entre ocho y doce horas.

size = 150	La dimensión de los vectores. Cuánto mayor es la dimensión, mejor es el rendimiento, hasta un momento (dimensión 300 aproximadamente) en que el rendimiento se estabiliza [48].
window = 5	La cantidad de palabras que se consideran en el contexto para entrenar los vectores.
min_count = 30	La cantidad de veces que tiene que aparecer en el corpus una palabra para que se le cree un vector.

CUADRO 4.2: Parámetros para la generación de los Embeddings

Al principio utilizamos tamaño 400 y min_count 1, pero los embeddings quedaron demasiado pesados y la plataforma muy lenta, ya que deben cargarse todos en memoria para hacer uso de los mismos. Además, Wikipedia contiene demasiadas palabras en otros idiomas (y de otros alfabetos), por lo que había muchas palabras inútiles convertidas en vector.
Para contrarrestar estas dos cosas, cambiamos el min_count a 30, y el tamaño a 150, como mencionamos antes.
No observamos un gran cambio en los resultados, pero sí una gran reducción en el tamaño de los embeddings, que pasaron de pesar 1GB a 500 MB. Para el tamaño de ventana, elegimos el que viene por defecto que es 5. No probamos con distintos tamaños por el tiempo que demora en entrenar.

4.2. Ejercicios

Como mencionamos previamente, la plataforma cuenta con cuatro tipos diferentes de ejercicios que pueden ser generados por parte de los docentes y posteriormente resueltos por los estudiantes. En un principio se propusieron muchos más que nos hubiera gustado elaborar pero, dado el alcance de este proyecto, decidimos quedarnos solamente con estos cuatro que eran considerablemente distintos entre sí y cada uno contaba con desafíos propios. Cada actividad hace uso de los distintos recursos construidos y de elementos del procesamiento de lenguaje natural, que fueron explicados previamente.

En esta sección pasamos a describir cada uno de estos ejercicios, contando qué es lo que hacen, sus funcionalidades, el proceso para llevarlos a cabo y las principales dificultades que se presentaron durante la implementación junto con las soluciones propuestas elegidas por el equipo.

4.2.1. Ejercicio de conjugación de verbos

Este ejercicio consiste en completar espacios en blanco de un texto con el verbo conjugado en el tiempo verbal correcto (indicado al lado de cada espacio en blanco) (ver figura 4.5). Fue el primero que implementamos, y el más simple de los cuatro desde el punto de vista de procesamiento de lenguaje.

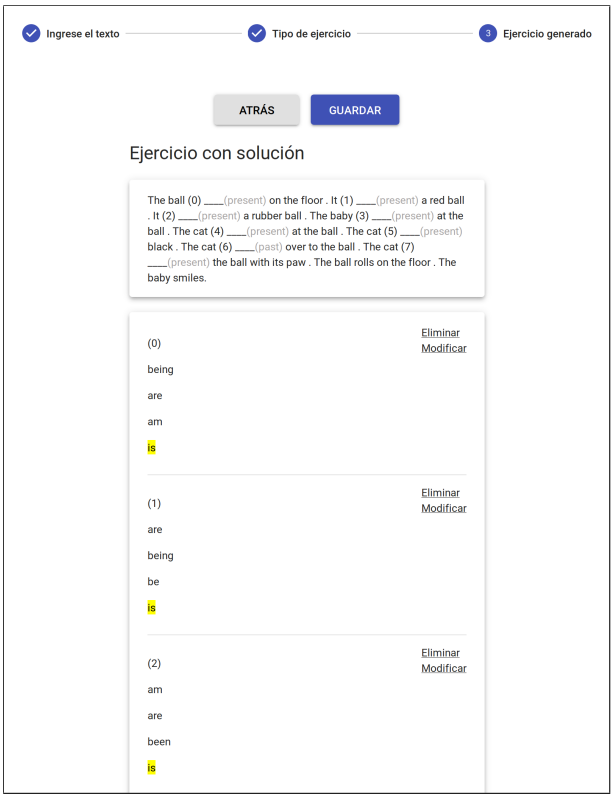

FIGURA 4.5: El ejercicio de conjugación de verbos desde el punto de vista del docente.

Funcionamiento y procesamiento

Primero se tokeniza el texto con el método "word_tokenize" de NLTK [14] y se taggea el texto separado por párrafos utilizando el POS tagger de Stanford [15], para poder reconocer los verbos (analizando los lemas de las palabras). Luego, se guardan los verbos en una lista y se eliminan aquellos que no aparezcan en el vocabulario generado. A partir de las etiquetas morfológicas, se obtiene el tiempo verbal del verbo original que luego es utilizado para mostrar en el ejercicio la sugerencia del tiempo verbal correcto.

Para generar los distractores, utilizamos la librería Pattern [42] que proporciona un método llamado "lexeme" que retorna todas las posibles conjugaciones de un verbo ingresado. Para cada una de las palabras se eligen tres

conjugaciones (diferentes a la conjugación original del verbo) que pasarán a ser los distractores.

En este ejercicio únicamente utilizamos el vocabulario de los recursos generados. Nuestro principal objetivo fue probar los taggers y las librerías, y realizar una prueba de concepto.

Pasos de procesamiento

1. Se separa el texto en párrafos (por saltos de línea).
2. Se tokeniza los párrafos con NLTK y se taggea el texto con el POS tagger de Stanford, para poder encontrar los verbos.
3. Se eliminan los verbos que no están en el vocabulario.
4. Para cada verbo, se generan sus conjugaciones con Pattern
5. Se seleccionan las tres conjugaciones más frecuentes como los distractores, diferentes de la original.
6. Se obtiene el tiempo verbal del verbo solución para utilizar como sugerencia en el ejercicio.

Problemas encontrados y modificaciones realizadas

Una variante que manejamos fue no mostrar el tiempo verbal, pero esto nos daba problemas. Por un lado, si se dejan las conjugaciones en muchos tiempos verbales, podría haber ambigüedad respecto a la opción correcta, pero, por otro lado, si se eliminan estas opciones, no quedan suficientes distractores para hacer el ejercicio viable.

Otro problema encontrado fue con los gerundios. Los verbos que son gerundios pueden estar antecedidos por otro verbo, por ejemplo ***He is going to the zoo*** tiene los verbos **is** y **going** juntos. Esto nos dio problemas al generar los ejercicios porque se tomaban los dos verbos contiguos y quedaban dos espacios en blanco. Por lo tanto, preferimos no seleccionar a los verbos gerundios como opciones. Para detectarlos y filtrarlos utilizamos la etiqueta gramatical dada por el POS tagger. Lo mismo ocurrió para los verbos conjugados en pasado y presente perfecto, por ejemplo en ***He had been to Uruguay*** y ***He has been to Uruguay*** y lo solucionamos de la misma manera.

La negación del verbo **"be"** o **"do"** también nos presentó un problema, porque el mismo puede ser expresado con contracciones, es decir, se puede escribir como ***is not*** o ***isn't***. Cuando la negación es contraída los POS taggers y parsers separan los tokens en dos diferentes, por lo que se pierde la referencia de la palabra. Como queríamos mantener el texto original ingresado, la solución fue no tomar los verbos negados que están contraídos y omitir las conjugaciones negadas como distractores. Para detectar los verbos negados contraídos utilizamos una expresión regular que verifica si la palabra finaliza con **"n't"**.

4.2.2. Ejercicio de sustantivos

El objetivo de este ejercicio es que el estudiante pueda aprender el significado de nuevas palabras. Es más complejo que el de conjugación de verbos ya que utilizamos varios recursos para su desarrollo.

Dado un texto, tomamos los sustantivos que aparecen en él y, según el vocabulario que el docente quiera trabajar, seleccionamos sustantivos y definiciones para que los estudiantes puedan escribir cada palabra con su definición correcta (ver figura 4.6).

Para este ejercicio utilizamos el diccionario de Wordsmyth, combinado con la lista de palabras Movers para asegurar que las palabras que se seleccionan no sean muy difíciles (o que no disten mucho del vocabulario a tratar). En este ejercicio el docente puede sustituir las definiciones por otras que considere más adecuadas, sin modificarlas en la base de datos.

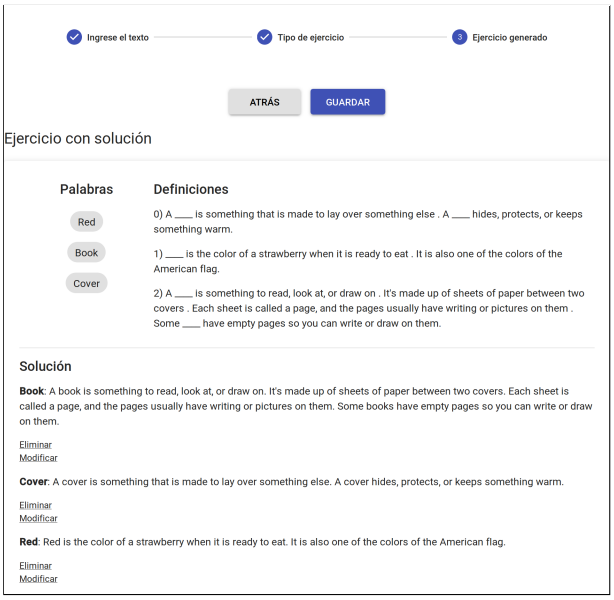

FIGURA 4.6: El ejercicio de sustantivos desde el punto de vista del docente.

Funcionamiento y procesamiento

Se tokeniza el texto ingresado con el método "word_tokenize" de NLTK [14] y se taggea el texto utilizando el POS tagger de Stanford [15]. Luego, a partir de las etiquetas gramaticales, se obtienen los sustantivos y removemos los que no se encuentren en el vocabulario.

Para encontrar la definición de cada sustantivo, se busca en nuestro diccionario generado, si existe una única definición entonces se devuelve, pero si existe más de una definición o no existe entonces se tiene que usar el algoritmo Lesk para buscar en WordNet [17] el Synset que tenga más sentido para el contexto. A pesar de que el algoritmo de Lesk no es del todo preciso, como se trata de textos y definiciones simples, funciona bastante bien. Por otro lado, si bien existen otras implementaciones para lograr la desambiguación del sentido de las palabras (WSD), la mayoría requieren un esfuerzo computacional que no lo amerita para este ejercicio, ya que la mayoría de las veces la definición es encontrada en el diccionario y además al ser una herramienta supervisada el docente tiene la oportunidad de corregir la definición encontrada con Lesk.

Pasos de procesamiento

1. Se tokeniza y se taggea el texto, para poder encontrar los sustantivos.
2. Se eliminan los sustantivos que no están en el diccionario.
3. Para cada sustantivo que tenga más de una definición, utilizamos Lesk para encontrar el synset correspondiente, y buscamos una definición cuya palabra contenga el mismo synset. De lo contrario se selecciona la definición encontrada en nuestro diccionario.

Problemas encontrados y modificaciones realizadas

En este ejercicio tuvimos problemas con las definiciones. Para empezar, casi todas contienen la palabra que están definiendo. Para solucionar este problema, decidimos ocultar la palabra de la definición. Por otro lado, el hecho de que contenga la palabra en la definición ayuda a que el estudiante aprenda cómo usarla además de su significado.

Otro problema fue que la lista de palabras es acotada: cuenta con 3567 definiciones. Por lo que agregamos la posibilidad de que los docentes agreguen nuevas palabras al diccionario.

Como en la aplicación de los estudiantes se tiene que escribir la palabra que corresponde con la definición, nos pareció adecuado utilizar la distancia de Levenshtein para alertar al estudiante cuando escribe la palabra con algún posible error de ortografía. Se muestra la alerta cuando la distancia de edición entre la respuesta del estudiante y la palabra correcta es exactamente igual a uno.

Posibles mejoras

Dado el nivel de inglés solicitado en los requisitos, el algoritmo de Lesk funciona bien. Pero, si se quiere extender la aplicación para soportar niveles de inglés más complejos, entonces es necesario sustituir el algoritmo de Lesk por algún método con mayor precisión como un clasificador Naive Bayes para desambiguar el synset de las palabras.

4.2.3. Ejercicio de uso del lenguaje

El ejercicio de uso del lenguaje (lo que vendría a ser equivalente a un ejercicio de Use of English en inglés) consiste en un texto con espacios en blanco, y se tiene que seleccionar en cado caso la palabra correcta de entre cuatro opciones, es decir, cada espacio en blanco cuenta con tres distractores y una opción correcta (ver figura 4.7).

Se llama ejercicio de uso del lenguaje porque diferentes palabras (desde el punto de vista gramatical) del texto pueden ser elegidas para que el estudiante las complete. Optamos por tomar como candidatos a los verbos, sustantivos y preposiciones del texto ingresado aunque puede ser fácilmente extendido para otras categorías gramaticales. Entonces, el objetivo de este ejercicio es que el estudiante aprenda a utilizar el lenguaje en un contexto dado, en nuestro caso, el texto recibido como entrada.

La mayor dificultad de este ejercicio se encuentra en la selección de los distractores. Al sustituir la palabra correcta por el distractor en la oración, se tiene que dar como resultado una oración semánticamente incorrecta, también podría dar una oración sintácticamente incorrecta, pero en ese caso sería muy fácil para el estudiante encontrar los distractores. Por otro lado, los distractores no pueden ser muy incoherentes con el contexto.

Para ejemplificar lo anterior tomamos la siguiente oración:
"***She ____ breakfast in the morning***"

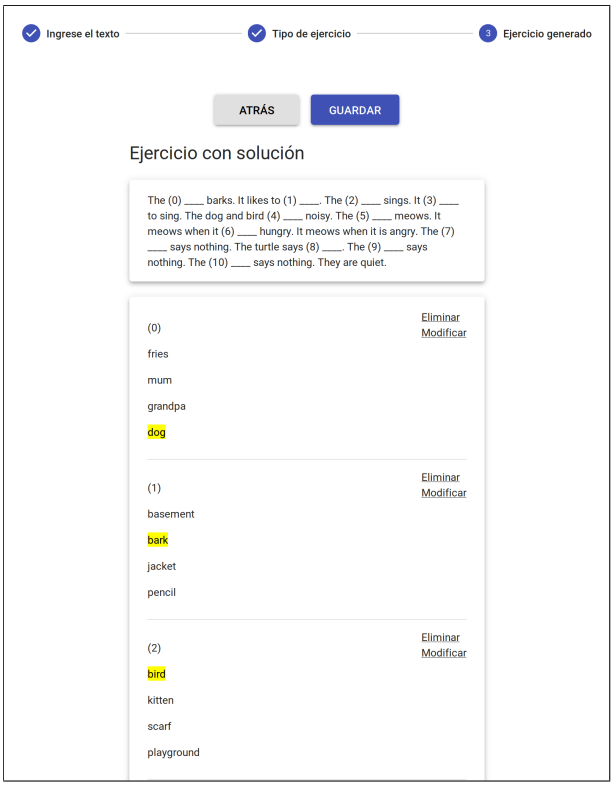

FIGURA 4.7: El ejercicio de uso del lenguaje desde el punto de vista del docente.

Donde la solución es ***has***. Un buen distractor sería ***runs*** porque semánticamente la oración ***She runs breakfast in the morning*** no tiene sentido, pero sintácticamente es correcta. Un mal distractor sería ***eats*** porque es semánticamente y sintácticamente correcta lo que causaría que dos opciones sean válidas.

Durante el desarrollo de este ejercicio fue que generamos los principales recursos como el modelo de lenguaje y de word embeddings. Estos fueron necesarios para poder resolver la problemática previamente pleanteada.

Funcionamiento y procesamiento

Como en el resto de los ejercicios, se comienza ingresando un texto, que es separado en oraciones con NLTK [14] para luego ser parseado y taggeado con el POS tagger de Stanford para obtener las palabras que según su categoría gramatical son candidatas para ser completadas por el estudiante. Para nuestro ejercicio de uso del lenguaje filtramos las categorías verbos, sustantivos y preposiciones.

Luego de obtener la lista de palabras posibles, se filtran aquellas que no pertenecen a nuestro vocabulario generado (ver sección 4.1.4) para asegurarnos que las palabras seleccionadas sean de interés educativo. Y por otro

lado, para mejorar los resultados por parte del modelo de lenguaje ya que del mismo se removieron los trigramas que no pertenecían al vocabulario para poder reducirlo.

Dada la oración *"The **cat sleeps under** the **table**"*, las palabras candidatas son ***cat***, ***sleeps***, ***under*** y ***table***, pues nos quedamos con los verbos, sustantivos y preposiciones. Sin embargo, en cada oración se va a tomar una sola palabra para ser completada por el estudiante. Una función elige de entre todas las palabras posibles la más frecuente. En el ejemplo planteado, la más frecuente según nuestra lista de palabras frecuentes (ver 4.1.3) es la palabra ***under***. Luego de obtenida la palabra correcta que será completada, el algoritmo procede a buscar los distractores para la misma.

Para determinar los distractores para cada opción utilizamos un algoritmo con estrategia de tipo embudo o filtros. Donde cada filtro tiene un propósito sobre la generación de distractores (ver figura 4.8).

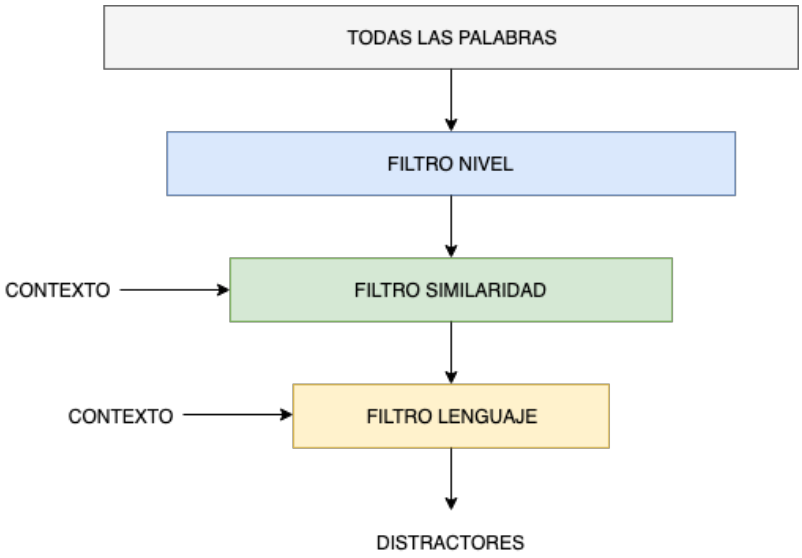

FIGURA 4.8: Estrategia de filtros para obtener distractores

El primero es el **filtro de nivel**, el objetivo de este filtro es lograr que todos los distractores estén comprendidos dentro del nivel de inglés adecuado para los estudiantes. Para lograr esto, el filtro utiliza la lista de palabras Movers y la etiqueta gramatical de la opción correcta para seleccionar solamente las palabras que tengan la misma categoría.

Uno de los problemas encontrados con el filtro de nivel de inglés, fue que el filtro no conjugaba los verbos en el tiempo verbal de la oración, lo que daba como resultado que la opción correcta esté en el tiempo verbal de la oración y el resto de los distractores no.

Para poder conjugar adecuadamente los distractores, primero se detecta si la opción correcta es un verbo y, en caso afirmativo, se obtiene el tiempo verbal utilizando la etiqueta gramatical dada por el POS tagger. Por último

se conjugan los verbos obtenidos a partir del filtro de nivel de inglés con el método que provee Pattern [42].

El segundo filtro es el **filtro de similitud** que es necesario para que los distractores sean similares a la opción correcta. Este filtro no sólo es importante para que el ejercicio sea desafiante, sino que también ayuda a evitar que se repitan palabras como opciones al acortar la lista de distractores según cada oración.

Este filtro espera como entrada los candidatos a distractores obtenidos por el filtro anterior y la opción correcta de la oración original. Para cada candidato, el modelo de word embeddings lo compara contra la opción correcta y devuelve una probabilidad de similitud. Luego de tener la lista de distractores asociados a la probabilidad anterior, se prosigue a filtrarlos según tres parámetros listados en el cuadro 4.3. Estos parámetros van a determinar la cantidad de distractores que se devuelven y qué tan similares a la opción original van a ser. Por defecto los parámetros ya tienen un valor asignado que está ajustado para que dada la cantidad de distractores que devuelve el primer filtro retorne una cantidad considerable para el siguiente filtro de lenguaje, pero pueden ser regulados en caso de que cambie la cantidad de palabras retornadas por el primer filtro o que se quiera aumentar el nivel de dificultad del ejercicio.

cota_similitud = 0.2	La cota de similitud determina cuales distractores serán filtrados. Los distractores que tengan una probabilidad de similitud menor a la cota serán descartados.
minimo_a_retornar = 90	Este parámetro fuerza a retornar un mínimo de distractores. En caso de que la cota de similitud devuelva una cantidad menor, entonces se continuará aumentando la lista con los siguientes distractores más similares hasta llegar al mínimo.
maximo_a_retornar = 350	Determina una cantidad máxima de distractores a retornar. Solo aplica en el caso de que la cantidad de distractores devueltos por la cota de similitud supere ese máximo.

CUADRO 4.3: Parámetros para el filtro de similitud

Por otro lado, un problema que nos surgió al ajustar los parámetros para que se retornen distractores muy similares es que en ocasiones el filtro dejaba pocos distractores para el siguiente filtro (el de modelo de lenguaje), haciendo que éste no fuera útil al retornar varias opciones correctas a la vez (cuando la funcionalidad principal del filtro de modelo de lenguaje es evitar que esto suceda).

También hay que ser cuidadoso al ajustar los parámetros para que no devuelva demasiados distractores, porque en ese caso el filtro no cumplirá con

su objetivo. Por esa razón agregamos el parámetro **maximo_a_retornar** que es la cantidad máxima de distractores que el filtro puede retornar.

En la sección de evaluación se muestran diferentes ejemplos con los resultados de este filtro cambiando los parámetros (ver 4.3.3). En particular, los resultados para los parámetros por defecto que elegimos.

Por último está el **filtro de lenguaje**. Su objetivo es poder seleccionar los distractores que son sintácticamente correctos pero semánticamente incorrectos. Sabemos que serán gramaticalmente correctos porque el filtro de nivel ya seleccionó únicamente distractores que tengan la misma categoría gramatical que la opción correcta.

El filtro recibe como parámetros la opción correcta, la oración original, la lista de distractores posibles dada por el filtro anterior y la cantidad de distractores que se quiere retornar. Este último por defecto es tres ya que se quiere tener cuatro opciones para el ejercicio.

La opción correcta, la oración original y la lista de distractores posibles son utilizados para generar una lista de oraciones donde se sustituye la opción correcta por cada distractor, es decir, que para cada distractor se genera una oración. Luego, con el modelo de lenguaje se le asigna una probabilidad a cada oración y se ordena la lista de peor a mejor según su probabilidad (las oraciones con peor probabilidad al comienzo). Y por último se devuelven los primeros distractores de la lista según el parámetro de cuantos se quieren retornar.

De esta forma, el filtro de lenguaje nos asegura (siempre y cuando tenga suficientes distractores) que la oración en la que se reemplaza la palabra por el distractor tendrá una muy baja probabilidad de ocurrencia en comparación con la oración original, lo que evita que se generen varias respuestas correctas.

Algoritmo filtros

Input: Palabra p, oración o
Output: Lista de distractores
$palabras_movers \leftarrow lista\ de\ palabras\ movers$
$categoria_palabra \leftarrow categoria\ gramatical\ p$
$distractores \leftarrow filtro_nivel(palabras_movers, categoria_palabra)$
$distractores \leftarrow filtro_similitud(distractores, p)$
$cantidad_distractores \leftarrow 3$
$distractores \leftarrow filtro_lenguaje(distractores, p, o, cantidad_distractores)$
return *distractores*

Pasos de procesamiento

1. Se separa el texto en oraciones con NLTK.
2. Para cada oración, se tokeniza y se taggea, para poder encontrar verbos, sustantivos y preposiciones.
3. De todas las palabras candidatas de la oración, se elige la más frecuente como la opción correcta a ser completada.
4. Para cada opción de una oración, se aplica el algoritmo de filtros ingresando la palabra, la oración y las etiquetas gramaticales.
5. Se retorna el texto con un conjunto de opciones con tres distractores asociados.

Problemas encontrados y modificaciones realizadas

En la generación de este ejercicio nos encontramos con diferentes problemas que no necesariamente están relacionados entre sí.

1. **Distractores frecuentes**

   Habían algunos distractores que siempre aparecían sin importar el texto ingresado e incluso se repetían varias veces en el mismo ejercicio. Esto sucedía porque en ese momento no contábamos con un buen modelo de word embeddings, por lo que el filtro de similitud no funcionaba de la manera esperada, y esto generaba que al filtrar por el modelo de lenguaje siempre las palabras menos frecuentes en el lenguaje eran muy buenas candidatas a ser distractores por la baja probabilidad de la oración.

   Hubo tres factores que impactaron en la solución de este problema. El primero fue que empezamos a considerar sólo las palabras dentro de nuestro vocabulario generado, lo que minimizó las palabras poco frecuentes como distractores. El segundo factor fue que mejoramos el modelo de word embeddings con un corpus más extenso (ver 4.1.7). Y por último, ajustamos mejor los parámetros del filtro de similitud para que nos retorne un conjunto más acotado de distractores.

2. **Distractores repetidos**

   Como los distractores están limitados por nuestro vocabulario generado, sucede que a veces algunos distractores están repetidos en un mismo ejercicio. Esto no es necesariamente malo, pero para mejorar la calidad de los ejercicios generados optamos por tomar sólo distractores que aún no fueron utilizados.

3. **Opciones correctas repetidas**

   Como mencionamos anteriormente, de todas las palabras candidatas para ser una opción a ser completada tomamos sólo una según su frecuencia. Esto genera el problema de que se repiten aquellas palabras muy frecuentes. Por ejemplo si tenemos varias oraciones con el verbo ***go***, que es muy frecuente, entonces siempre se tomará como la mejor opción.

   La solución fue modificar la forma en que elegimos a la mejor palabra de las candidatas dentro de una oración. Si todas las palabras están disponibles, es decir, que no fueron utilizadas previamente, entonces se retorna la más frecuente. Pero en caso contrario se retorna la siguiente en el orden de frecuencia que no fue utilizada.

4. **Resultados erróneos del modelo de lenguaje**

   Una de las observaciones del modelo de lenguaje generado fue que el corpus utilizado carecía de signos de puntuación (ver 4.1.6).

   Nos dimos cuenta de que cuando obteníamos la probabilidad de la oración en el filtro de lenguaje no quitábamos los signos de puntuación. Y como el modelo de lenguaje fue entrenado sin signos de puntuación no reconoce a estos caracteres, por lo que baja la probabilidad a la oración. Decidimos entonces procesar la oración previamente a analizarla con el modelo de lenguaje quitando todos los signos de puntuación.

5. **Distractores incoherentes**

   Al comienzo del desarrollo, cuando solo contábamos con el filtro de lenguaje, nos encontramos con distractores incoherentes. Los distractores que consideramos incoherentes son aquellos que no están dentro del vocabulario esperado del estudiante o que son demasiado distantes del contexto dado. En el ejemplo ***"She has breakfast in the morning"***, un distractor incoherente para la palabra ***has*** podría ser ***freezes*** ya que, por más de que cumple con las características de un buen distractor (correcto sintácticamente pero incorrecto semánticamente) esta muy alejado del contexto, lo que causaría que fuese demasiado sencillo para el estudiante encontrar la opción correcta. La solución a este problema fue el filtro de niveles, que elimina palabras fuera del vocabulario y el filtro de similitud para que los distractores estén dentro del contexto de la oración.

6. **Falta de pruebas**

   Si bien encontramos muchos errores probando manualmente la generación de ejercicios, el principal problema que tuvimos en este ejercicio, y, en cierta medida, en todos, fue la dificultad de realizar pruebas.

   La calidad de los ejercicios es difícil de medir con métodos cuantitativos, dado que es un tema subjetivo. Esto nos dificultó (entre otras cosas) el ajuste de parámetros, aunque más adelante en el informe mostramos cómo se comportan estos parámetros para diferentes textos ingresados.

Posibles mejoras

En caso de querer abarcar más uso del lenguaje en el ejercicio, se podría extender la funcionalidad que toma las categorías gramaticales para obtener más palabras candidatas. Nosotros consideramos que tomando las categorías verbos, sustantivos y preposiciones era suficiente para el nivel planteado en los requisitos. También sería de utilidad que el docente pueda seleccionar cuáles categorías considerar previamente a generar el ejercicio.

Como indicamos en los problemas encontrados, la falta de pruebas cuantitativas nos limitó para poder determinar la calidad de los ejercicios generados. Si se pretende extender esta herramienta para niveles de inglés más avanzados, se podría crear un clasificador entrenado con ejercicios de uso del lenguaje ya anotados por docentes para luego utilizarlo en nuestra herramienta como una línea base para determinar la calidad de los ejercicios.

4.2.4. Ejercicio de hipónimos

El ejercicio de hipónimos fue el último que implementamos y lo hicimos con el fin de comenzar a investigar este tipo de estrategia para generar ejercicios.

El objetivo principal es aprender a relacionar y comprender las palabras del lenguaje. Y consiste en que el estudiante pueda agrupar las palabras encontradas en el texto en diferentes categorías (ver figura 4.9).

La plataforma ya cuenta con determinadas categorías definidas que pueden ser extendidas. El docente ingresa un texto y se obtienen todas las palabras que son hipónimos de alguna de las categorías. Luego, el estudiante indica a cuál pertenece cada palabra.

El principal recurso utilizado en este ejercicio fue la base de datos léxica WordNet [16], ya que permite encontrar relaciones entre palabras, en particular, la relación de hiponimia.

FIGURA 4.9: El ejercicio de hipónimos desde el punto de vista del docente.

Funcionamiento y procesamiento

Todo el texto ingresado es tokenizado y taggeado con el POS tagger de Stanford. Luego, se filtran sólo las palabras que son sustantivos o adjetivos utilizando su categoría gramatical.

Una vez obtenidas todas las palabras que van a ser analizadas, para cada palabra candidata se comprueba si es un hipónimo de alguna de las categorías. Esta verificación se hace con un algoritmo simple que busca si existe la relación de hiponimia entre la palabra y la categoría.

El algoritmo utiliza WordNet para buscar las relaciones entre palabras. Toma como entrada el synset de la categoría y el de la palabra. El synset de la categoría ya está definido en nuestra base de datos, pero el desafió de este ejercicio fue encontrar el synset de la palabra que lleve al algoritmo por el camino correcto para encontrar su hiperónimo. Una alternativa posible fue tomar todos los synsets y buscar alguno que tuviera una categoría de hipónimo, pero esto aumentó el tiempo de procesamiento en forma exponencial, ya que el algoritmo recorría varios niveles de profundidad en las relaciones para cada synset. Después de investigar y probar diferentes opciones, optamos por elegir el primero de los synsets de la palabra. Con el synset de la palabra solo se necesita recorrer sus synsets hiperónimos de forma recursiva para verificar si alguno coincide con los synsets de las categorías.

Si no se cuenta con una condición de parada el algoritmo podría llegar

a demorar mucho tiempo, ya que recorrerá hasta llegar al synset base. Por eso se define una cota que determina el nivel de profundidad que es posible alcanzar. La cota es definida por defecto en veinte, es decir, que si la relación se encuentra a una distancia mayor a veinte entonces el algoritmo nunca llega a encontrarla.

Algoritmo recursivo hipónimos

```
Input: synset_palabra, synset_categoria, cota = 20
Output: Si existe la relacion hipónimo-hiperónimo entre los synsets
hiperónimos = synset_palabra.hypernyms()
if es_vacio(hiperónimos) then
|   return False
end
if cota = 0 then
|   return False
else
|   es_hiperónimo = False
|   foreach hiperónimo de los hiperónimos do
|   |   if hiperónimo = synset_categoria then
|   |   |   es_hiperónimo = True
|   |   else
|   |   |   es_hiperónimo = algoritmo_hipónimos(hiperónimo,
|   |   |     synset_categoria, cota - 1)
|   |   end
|   end
|   return es_hiperónimo
end
```

Pasos de procesamiento

1. Se tokeniza y se taggea el texto, para poder encontrar sustantivos y adjetivos.
2. Para cada palabra, se busca si es hipónimo de alguna de las categorías usando el algoritmo de hipónimos.
3. Si se encuentra la relación de hiponimia entre la palabra y la categoría, se retorna como una palabra para unir en el ejercicio.

Problemas encontrados y modificaciones realizadas

Como este ejercicio fue experimental, encontramos algunos problemas que no fueron resueltos y se plantean como trabajos a futuro.

El principal problema no resuelto de este ejercicio es que los textos generalmente contienen muchas palabras de una misma categoría, es decir que no es común que un texto abarque muchas categorías diferentes. Esto implica que el ejercicio puede devolver muchas palabras pertenecientes a la misma categoría (por ejemplo: si el texto habla de animales entonces es probable que la mayoría de sus palabras sean de la categoría animales). Como las categorías son acotadas, también puede suceder que no se encuentren suficientes palabras en el texto para que el ejercicio tenga sentido.

En un principio solo tomábamos los sustantivos como hipónimos, pero observamos que algunas palabras que queríamos utilizar como hipónimos no se tenían en cuenta, en particular los colores. Los colores generalmente son usados como adjetivos (por ejemplo ***"The red car"***). Como el texto ingresado no es mostrado en el resultado final del ejercicio, realmente no importa cual es la categoría gramatical o el sentido que tiene la palabra en el texto, nuestra primera aproximación fue tomar sustantivos ya que analizar todas las palabras sería un proceso muy costoso en procesamiento. Decidimos entonces trabajar sobre los sustantivos y también los adjetivos, lo que solucionó muy bien el problema.

Otro problema fue que el algoritmo que verifica la relación de hiponimia toma como entrada un solo synset, cuando es probable que una palabra tenga varios synsets asociados. Una opción que descartamos fue analizar todos los synsets, por eso probamos diferentes alternativas.

En el primer intento probamos con el algoritmo de Lesk pero no obtuvimos muy buenos resultados ya que el contexto condiciona demasiado el synset a elegir y en este ejercicio no importa el contexto en que aparece la palabra. En general, las categorías aparecían asociadas al synset más común.

Luego probamos con obtener el synset más frecuente. El funcionamiento es tomar cada lema del synset, contar cuántas veces aparece en un corpus (en nuestro caso usamos el corpus Brown [31]) y quedarse con el synset con mayor número de ocurrencias. Aunque parecía prometedor nos encontramos con algunos casos donde fallaba, un ejemplo fue con la palabra **cat** donde se devolvía **'guy.n.01'**, cuya definición es ***"an informal term for a youth or man"***, como el synset más frecuente.

Finalmente nos decidimos por una opción más fácil y rápida de implementar que las anteriores pero que funciona mejor para nuestro caso, que fue tomar el primer synset. Como las categorías que tenemos son básicas y siempre se esperan synsets de sustantivos, entonces esta solución funciona porque el primer synset siempre es un sustantivo y para la mayoría de los

casos se corresponde con el sentido más frecuente dentro del vocabulario. No ocurre en la totalidad de los casos ya que por ejemplo el primer synset de "planta" devuelve "planta industrial" en lugar de "vegetal". El motivo de esto se encuentra en que WordNet está basado en un corpus que contiene muchas noticias de negocios.

Posibles mejoras

Como vimos que los textos ingresados usualmente no aplican para generar un ejercicio con suficientes palabras de distintas categorías, se podría utilizar el procesamiento de este ejercicio para combinarlo con algún otro.

Una idea que se nos ocurrió fue hacer un ejercicio de verdadero y falso, donde algunas de las afirmaciones sean formuladas con este procedimiento. Por ejemplo si encontramos que ***car*** es un hipónimo de ***vehicle*** podríamos afirmar que la sentencia ***"A car is a vehicle"*** es verdadera. Y de la misma manera podemos hacer afirmaciones que sean falsas con categorías que el procedimiento sabe que no son válidas, como la sentencia ***"A car is an animal"***.

Otra mejora posible es extender las categorías con las que contamos actualmente y que además el docente pueda seleccionar sólo aquellas que quiera considerar al momento de generar el ejercicio.

Sobre el problema de elegir un synset para una palabra, otra posible mejora que se podría probar es tomar todos los synsets pero filtrar por categoría gramatical (en nuestro caso los sustantivos), esto aumentaría el tiempo de respuesta porque se recorrerían más caminos para encontrar hiperónimos pero daría mejores resultados y se encontrarían más palabras.

4.3. Evaluación

En esta sección presentamos el resultado de la experimentación y evaluación de los recursos generados y los algoritmos. La mayoría de la experimentación fue empírica, buscando contra ejemplos que nos permitan detectar errores entre dos o más alternativas.

4.3.1. Modelo de lenguaje

Para el modelo de lenguaje probamos tres alternativas distintas (ver sección 4.1.6):

- Modelo entrenado con corpus Wikipedia Simple

- Modelo entrenado con corpus de Wikipedia
- Modelo entrenado con corpus de Wikipedia y reducido mediante un filtro de vocabulario

Fuimos evaluando el rendimiento de los diferentes modelos en la práctica a medida que implementábamos ejercicios. Los tres modelos mostraron un comportamiento similar en tiempo de procesamiento (todos demoraban menos de un segundo en devolver la probabilidad). Por lo tanto, lo que comparamos en esta sección es el resultado de las probabilidades.

Para evaluar los modelos tomamos una oración correcta y generamos variantes semánticamente incorrectas cambiando una de las palabras. Si el modelo funciona bien, la oración original debería tener una mejor probabilidad que el resto.

Comparamos la oración ***"he climbs the mountain"*** con otras variantes generadas a partir de cambiar el sustantivo **mountain** y obtuvimos los resultados mostrados en la tabla 4.4.

Oración	**Wikipedia Simple**	**Wikipedia**	**Wikipedia reducido**
he climbs the **mountain**	-15.76	-13.44	-13.44
he climbs the **dog**	-15.23	-16.26	-16.26
he climbs the **car**	-15.61	-16.46	-16.46
he climbs the **food**	-15.91	-17.19	-17.19
he climbs the **koala**	-16.32	-17.10	-22.05

CUADRO 4.4: Comparación de la probabilidad calculada con modelos de lenguaje para diferentes variantes de una oración

Los resultados del modelo de lenguaje entrenado con el corpus de Wikipedia Simple no fueron buenos porque la oración que tendría que dar más probabilidad tiene una probabilidad menor que algunas otras que son semánticamente incorrectas.

Por otro lado, en los modelos entrenados con el corpus de Wikipedia se aprecia una gran diferencia de probabilidades entre la oración original y el resto. Este es el resultado esperado. Como en el modelo de lenguaje reducido se eliminaron los trigramas que no pertenecían al vocabulario, en las primeras cuatro oraciones las probabilidades son las mismas que en el modelo no reducido, pero en la quinta es diferente ya que utiliza la palabra ***"koala"*** que no está contenida en el vocabulario. Se puede deducir entonces que el modelo reducido no funcionará bien con palabras que no aparezcan en el vocabulario, pero esto no es un problema en la generación de ejercicios porque siempre se escogen palabras que sí pertenecen al vocabulario.

4.3.2. POS tagging

En la sección 2.2.2, donde se presentó la tarea de POS tagging, se indicó que el POS tagger de Stanford daba, en algunos casos, mejores resultados que el de NLTK. Pero en esta sección se muestran los tiempos que tarda cada uno en lograr devolver las etiquetas gramaticales para una oración dada. El tiempo de procesamiento del POS tagging es importante ya que todos los ejercicios lo usan y los textos ingresados generalmente contienen varias oraciones.

Tomamos dos oraciones para comparar el tiempo y las etiquetas gramaticales que se devuelven (ver tabla 4.5).

Oración	**NLTK**	**Stanford**
The cat jumps over the table	[DT, NN, VBZ, IN, DT, NN]	[DT, NN, VBZ, IN, DT, NN]
	Tiempo: 0.002s	Tiempo: 0.901s
The quick brown fox runs fast	[DT, JJ, NN, NN, VBZ, RB]	[DT, JJ, JJ, NN, VBZ, RB]
	Tiempo: 0.003s	Tiempo: 0.9s

CUADRO 4.5: Comparación de los tiempos de respuesta de los pos taggers

Existe una diferencia de tiempo considerable entre un POS tagger y otro. El POS tagger de NLTK es mucho más rápido que el de Stanford. Sin embargo, en la segunda oración el POS tagger de NLTK falla al devolver las etiquetas, ya que la palabra **brown** debería tener la etiqueta JJ que corresponde a un adjetivo y tiene la de un sustantivo.

En general, encontramos que el POS tagger de Stanford devuelve mejores resultados que el de NLTK. Como el tiempo de procesamiento es importante, nuestra plataforma permite cambiar entre POS taggers con sólo una variable, esto logra que se ajuste según el contexto y las necesidades.

4.3.3. Ejercicio uso de lenguaje

Filtro de similitud

Evaluamos dos aspectos del filtro de similitud: el primero fue que se retornen suficientes palabras para el filtro de lenguaje (siguiente filtro a aplicar) y el segundo que el conjunto de palabras quede condicionado por la palabra correcta.

El principal recurso utilizado por el filtro de similitud es el modelo de word embeddings entrenado con el corpus de Wikipedia y ajustado como se muestra en la sección de la generación de word embeddings 4.1.7.

Uno de los parámetros que recibe el filtro de similitud es un valor que indica al cantidad mínima de palabras a retornar. También recibe como entrada la lista de palabras Movers para una categoría gramatical específica (filtro de niveles); por lo tanto, el valor de la cantidad mínima de palabras a retornar debe ser menor al total de palabras ingresadas al filtro, de lo contrario no se filtrará ninguna palabra.

En el cuadro 4.6 se muestra la cantidad total de palabras para las categorías gramaticales que se tienen en cuenta en este ejercicio. En el caso de las preposiciones no interesa que sean filtradas por la similitud ya que de por sí son muy pocas. El mínimo a retornar en el filtro de similitud debe ser menor a 122 palabras porque si es mayor o igual los verbos no serán filtrados. También se deduce que el parámetro que determina la cantidad máxima de palabras a retornar debe ser menor a 424.

Categoría gramatical	**Cantidad**
Sustantivos	424
Verbos	122
Preposiciones	26

CUADRO 4.6: Cantidad de palabras por categoría gramatical a partir del filtro de nivel

Los valores que nosotros elegimos para los parámetros de mínimo a retornar y máximo a retornar fueron **90** y **350** respectivamente.

Para evaluar si el filtro está condicionando el resultado según la palabra correcta, en la tabla 4.7 tomamos tres sustantivos diferentes como la opción correcta y le aplicamos el filtro de similitud a los sustantivos que toma como entrada. Mostramos la cantidad de palabras que devuelve el filtro.

Palabra	**Cantidad de distractores**	**Más similares**
dog	**186** de 424	cat, puppy, kitten
tree	**131** de 424	pear, leaf, flower
book	**90** de 424	film, letter, song, page

CUADRO 4.7: Cantidad de distractores para sustantivos

Si bien la cantidad de distractores es diferente para cada palabra, hicimos la intersección para saber si realmente están condicionados. Obtuvimos los siguientes resultados:

- $distractores('dog') \cap distractores('tree') = 87$
- $distractores('dog') \cap distractores('book') = 48$

- $distractores('tree') \cap distractores('book') = 30$

La salida del filtro difiere mucho según la palabra, lo que implica que logra satisfactoriamente que los distractores queden condicionados por la palabra correcta.

Además, para la palabra **"book"** notamos que el parámetro de mínimo a retornar se está aplicando. Todas las palabras tienen una salida lo suficientemente grande como para que el filtro de lenguaje pueda aplicarse.

Otro de los parámetros que ajustamos fue el de cota de similitud. Los distractores que tengan una probabilidad de similitud menor a la cota serán descartados. El valor por defecto que le asignamos a este parámetro fue **0.2**.

En el cuadro 4.7, con excepción de la palabra **book**, el resto esta siendo limitado por la cota de similitud ya que se está retornando una cantidad mayor al mínimo (90 distractores) pero menor al máximo (350 distractores).

Para evaluar la cota, cambiamos el valor de la misma a **0.3** y ajustamos el valor del parámetro que determina la cantidad mínima a retornar a 0 para poder ver cuántos está filtrando la cota. Obtuvimos en el cuadro 4.8 que la cantidad de palabras retornada es muy baja para lo que necesita el siguiente filtro. Por lo tanto, se deduce que el valor de este parámetro, para nuestro caso, debe ser menor a **0.3**.

Palabra	**Cantidad de distractores**
dog	**81** de 424
tree	**12** de 424
book	**38** de 424

CUADRO 4.8: Cantidad de palabras retornadas por el filtro con cota de similitud igual a 0.3 y mínimos a retornar igual a 0

Las intersecciones de distractores cuando la cota tiene valor **0.3** son mejores porque menor cantidad de palabras coinciden.

- $distractores('dog') \cap distractores('tree') = 10$
- $distractores('dog') \cap distractores('book') = 0$
- $distractores('tree') \cap distractores('book') = 0$

Filtro de lenguaje

El filtro de lenguaje devuelve los tres distractores finales asociados a la palabra elegida dentro de cada oración. Evaluar el filtro de lenguaje es análogo a evaluar el modelo de lenguaje, ya que la función del filtro de lenguaje

es ordenar según la probabilidad que el modelo de lenguaje calcula para la oración.

Se puede evaluar si llegan suficientes palabras al filtro de lenguaje para que la diferencia entre la palabra correcta y los distractores sea significativa. Entonces, evaluamos como se comporta una oración al pasar por todos los filtros, en particular vimos cuántos distractores recibe el filtro de lenguaje y las diferencias de probabilidades entre la oración original y las oraciones con los distractores.

Tomamos la oración ***"The boy runs fast in the garden"*** con la palabra **boy** como la opción correcta y analizamos la diferencia de probabilidad entre la oración con la palabra correcta y con la primer palabra de la lista ordenada que el filtro de lenguaje devuelve. El filtro de lenguaje recibe como entrada **147 distractores** como resultado de los filtros anteriores, luego ordena y decide que la palabra con menor probabilidad es **grandpa**. Si sustituimos **grandpa** por **boy** en la oración, la probabilidad según el modelo de lenguaje es de **-24,46**, mientras que la probabilidad de la oración original es de **-18,18**.

Consideramos que la diferencia entre la palabra correcta y los distractores es suficiente para que la oración con los distractores sea semánticamente diferente a la oración original.

Tiempos de respuesta

El ejercicio de uso de lenguaje utiliza varios recursos que consumen mucho procesamiento y tenemos que uno de los requisitos es que la plataforma sea utilizada en un contexto educativo, por lo tanto los tiempos de respuesta deben ser aceptables.

Usamos como muestra un texto con 26 oraciones que genera 21 opciones para completar. El texto de muestra representa un texto de largo promedio, ya que si el tamaño es mucho mayor no es práctico para los alumnos tener que completar un ejercicio con demasiadas opciones. Así que usamos estos tiempos como una cota superior para medir a nuestra plataforma.

> It was night time. He went out to walk his dog. His dog was big. The dog's name was Max. "Let's go, Max!" he said to his dog. He walked past his neighbor's house. He saw there were no lights in the house. His neighbor was not home. He walked past the market. There were people shopping for groceries. He remembered he needed milk. He tied Max to a pole. "Wait here, Max," he said. He walked inside the market. It took fifteen minutes to buy the milk. He untied Max and continued walking. He walked back to his house. He let Max inside the house. It was time to feed Max. He served Max a bowl of dog food. Max ate it quickly. It was late. He took Max to the backyard. Max slept in a dog house. He said good night to Max. Max barked.

En el siguiente cuadro 4.9 se muestran los tiempos de procesamiento para el ejercicio de uso del lenguaje con el texto de muestra. Se compara también con el POS tagger de NLTK y Stanford ya que demostramos anteriormente que impactan en el procesamiento.

Procesamiento	**Con NLTK**	**Con Stanford**
Selección opciones	3,050s	88,184s
Filtro niveles	0,015s	0,032s
Filtro similitud	0,142s	0,219s
Filtro lenguaje	0.159s	0,204s

CUADRO 4.9: Métricas de tiempos de la generación del ejercicio de uso del lenguaje

La selección de opciones utiliza principalmente el POS tagger, por eso es que los tiempos difieren tanto según el POS tagger que este activo. Además el mayor tiempo de procesamiento se encuentra en ese paso. Pudimos ver que el resto de los filtros tienen tiempos prácticamente insignificantes.

4.3.4. Ejercicio hipónimos

En el ejercicio de hipónimos la mayor complejidad computacional está en el algoritmo recursivo que determina si una palabra es hipónimo de una categoría.

Uno de los problemas que nos encontramos fue la selección de un synset para una palabra, ya que dependiendo del que encuentra el algoritmo, puede determinar si una categoría es hiperónimo o no. Como se mencionó en la sección 4.2.4, la estrategia que utilizamos fue tomar el primer synset de la palabra. Además comentamos que existe otra alternativa que nos asegura encontrar si existe la relación de hiponimia, pero que es demasiado costosa (aplicar el mismo algoritmo en cada uno de los synsets de la palabra, y obtener todas las categorías encontradas).

En el cuadro 4.10 evaluamos con algunos hipónimos pertenecientes a las categorías que la plataforma soporta, si son encontrados usando nuestra estrategia y comparamos los tiempos de procesamiento contra la estrategia de buscar en todos los synsets.

Palabra	**Estrategia primer synset**	**Estrategia todos los synsets**
green	colors - tiempo: 0,0023s	colors - tiempo: 1,09s
scarf	clothing - tiempo: 0,0021s	clothing - tiempo: 0,96s
boat	vehicles - tiempo: 0,013s	vehicles - tiempo: 1,287s
football	sport - tiempo: 0.015s	sport - tiempo: 1,24s
white	No encuentra - tiempo: 0.025s	color, clothing - tiempo: 1,54s

CUADRO 4.10: Comparación de estrategias para aplicar el algoritmo de hipónimos

Hay una diferencia considerable en tiempos entre una estrategia y otra. Pero en el caso de la palabra **white** la estrategia del primer synset no encuentra que es un hipónimo de color y la segunda estrategia si. Por lo tanto, si se cuenta con una buena capacidad de procesamiento la estrategia de considerar todos los synsets es mejor.

El algoritmo tiene un parámetro que define la profundidad en que se va a recorrer por los synsets hasta encontrar el hiperónimo. Si se supera esa profundidad el algoritmo asume que no existe la relación de hiponimia. El parámetro tiene un valor por defecto de 20. Para determinar ese valor evaluamos en el cuadro 4.11 algunas palabras de las categorías para ver cuál era el promedio de profundidad en dónde se alcanza la relación de hiponimia.

Palabra	**Distancia**
green	2
scarf	2
boat	3
football	2
tie	3

CUADRO 4.11: Evaluación de distancia entre la palabra y su categoría

En nuestro caso, al tratarse de categorías básicas la mayoría de las palabras que analizamos no superó la profundidad de 4 en el recorrido desde la palabra hasta la categoría. Nuestro parámetro tiene el valor de 20 para cubrir posibles casos bordes, pero podría ser mucho menor.

5. Aplicación y Arquitectura

Como ya mencionamos, la plataforma tiene dos interfaces, una orientada para el uso de los docentes y otra para el de los estudiantes, la idea de este capítulo es explicar ambas en detalle, además de la arquitectura y la distribución de la plataforma.

5.1. Aplicación

5.1.1. Interfaz de docentes

La interfaz de docentes es la más compleja ya que contiene todas las funcionalidades para generar los ejercicios que el alumno va a utilizar. Además, requiere de una conexión a internet constante porque en todo momento se conecta con el servidor. Esto no nos pareció una limitante ya que el docente puede crear los ejercicios en cualquier momento, y no necesariamente en el aula.

A continuación explicamos las funcionalidades con las que cuentan los docentes. Las mismas se pueden encontrar en el menú de navegación que se muestra en la figura 5.1.

FIGURA 5.1: Las opciones de la interfaz del docente.

Inicio / Textos

En la página de inicio se muestra la principal funcionalidad de la plataforma que es la generación de ejercicios (ver 5.2). Todos los ejercicios se generan siguiendo tres simples pasos:

1. Ingresar un texto en inglés

2. Seleccionar el tipo de ejercicio deseado
3. Modificar y generar ejercicio para luego guardarlo

Primero se le pide al docente que ingrese un texto en inglés en una entrada de texto plano. Si el docente lo desea puede utilizar textos de la base de datos seleccionando la pestaña Textos (Ver 5.4), allí se pueden ver todos los textos que los docentes alguna vez emplearon para generar ejercicios, ya que son almacenados junto con el ejercicio.

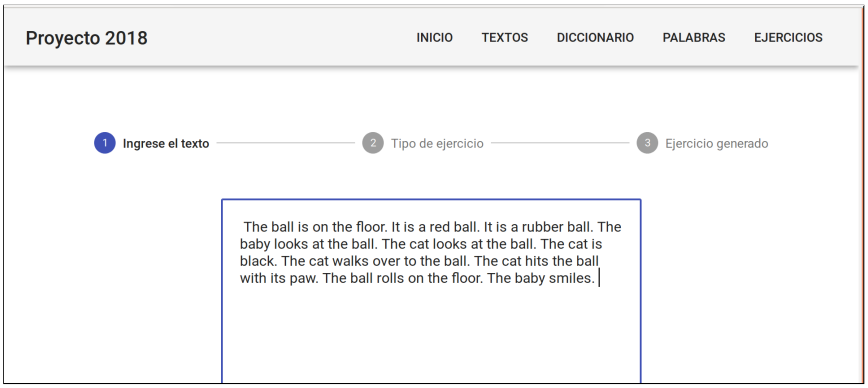

FIGURA 5.2: Ventana para que el maestro inserte el texto.

Una vez que el docente ingresa el texto con el que quiere trabajar, se lo conduce al siguiente paso que es seleccionar el tipo de ejercicio, como se puede ver en la figura 5.3.

FIGURA 5.3: Ejercicios que el docente puede generar.

Luego de que el docente decide qué ejercicio crear, se le muestra en pantalla el ejercicio generado. Además se le da la opción de modificar los distractores que crea pertinente, eliminar opciones o modificar definiciones según el tipo de ejercicio lo habilite, como se aprecia en la figura 4.7.

Una vez que termina de modificar y el ejercicio está listo para ser utilizado por los estudiantes, se le da la opción de guardar, lo cual persiste y deja

al ejercicio accesible para todos los alumnos. En caso de querer usar la actividad sin conexión a internet con los estudiantes, se brinda la posibilidad de descargar el ejercicio para luego pasárselo a cada uno de los alumnos.

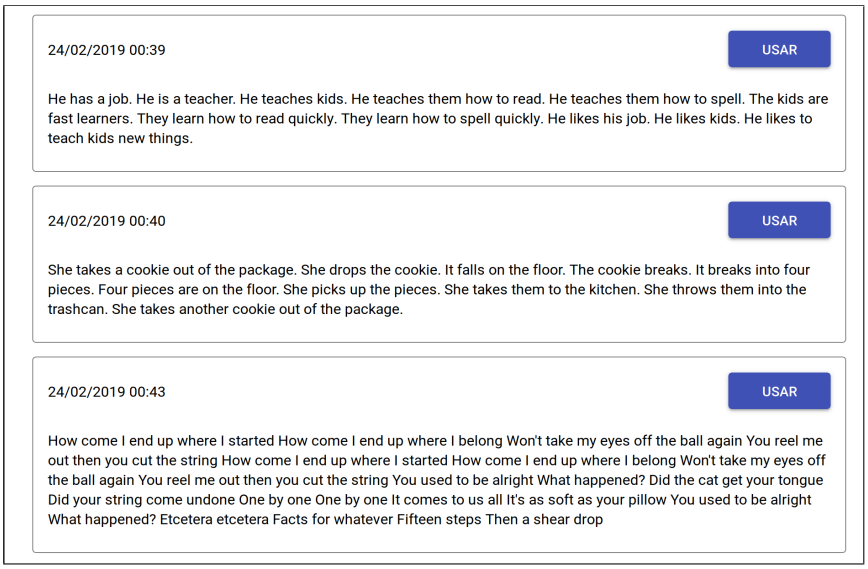

FIGURA 5.4: Lista de textos guardados en el servidor.

Diccionario

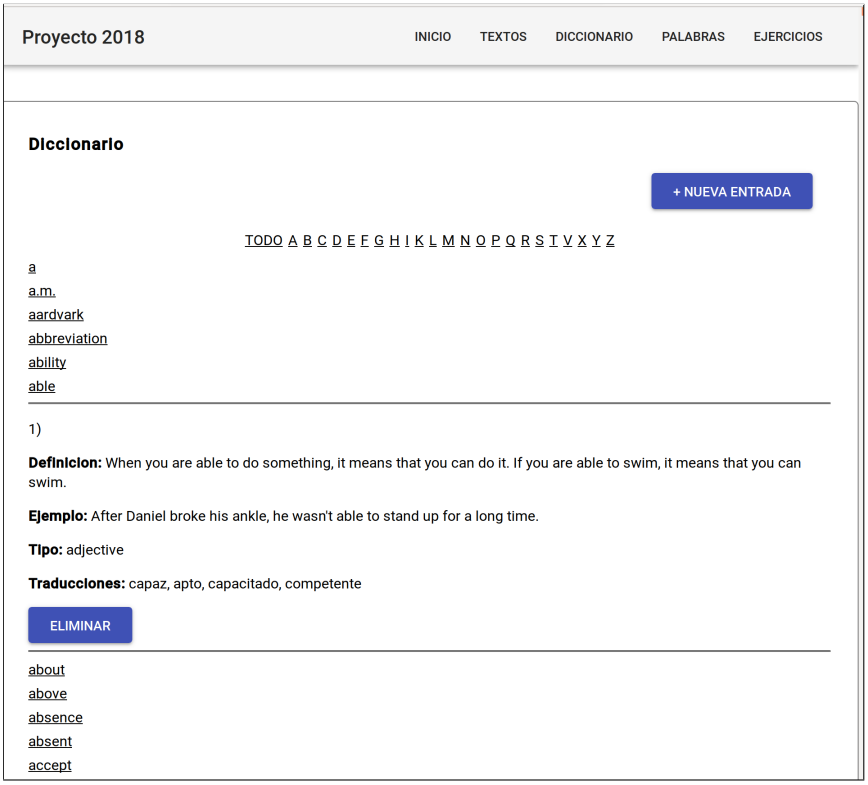

FIGURA 5.5: El diccionario de la base de datos.

El diccionario se utiliza para el ejercicio de sustantivos, y nos pareció importante permitir que el docente agregue y modifique las definiciones que crea conveniente.

En esta pestaña el docente puede ver las palabras que contienen definiciones, modificar o eliminar las que estén equivocadas y agregar nuevas, como se ve en la figura 5.5.

El problema que esto presenta es que las definiciones son compartidas por todos los usuarios, por lo que los docentes deberían manejar esta sección con responsabilidad para armar un diccionario que todos puedan usar.

De todas formas, si el docente quiere cambiar alguna definición sólo para un ejercicio específico siempre tiene la posibilidad de hacerlo al momento que se genera el ejercicio. Los cambios que se hacen durante la generación del ejercicio no modifican el diccionario.

Palabras

De forma similar a la sección del diccionario, en esta pestaña se puede visualizar el vocabulario de la aplicación. Este estuvo formado inicialmente por la unión de la lista de palabras Starters y Movers, las cinco mil palabras más frecuentes de la lista de palabras frecuentes y la conjugación de todos los verbos encontrados en las palabras Starters y Movers. Todos los ejercicios utilizan el vocabulario para determinar qué palabras pueden ser seleccionadas, por lo que agregamos esta opción para que se puedan agregar palabras más avanzadas o simplemente extender el vocabulario a otros niveles o tópicos.

Ejercicios

En esta sección los docentes pueden visualizar una lista ordenada por fecha de creación de los ejercicios ya generados, numerados por el atributo de id, e identificados por tipo de ejercicio como se puede ver en la figura 5.6.

Sobre cada ejercicio en la lista, se muestra un botón para eliminarlo en caso de que el docente crea que es necesario.

5.1.2. Interfaz de alumnos

La interfaz de los alumnos tiene menos funcionalidades que la del docente. En ella se muestra una lista con todos los ejercicios generados, y el alumno es capaz de seleccionar uno, resolverlo y corregirlo (ver la figura 5.7).

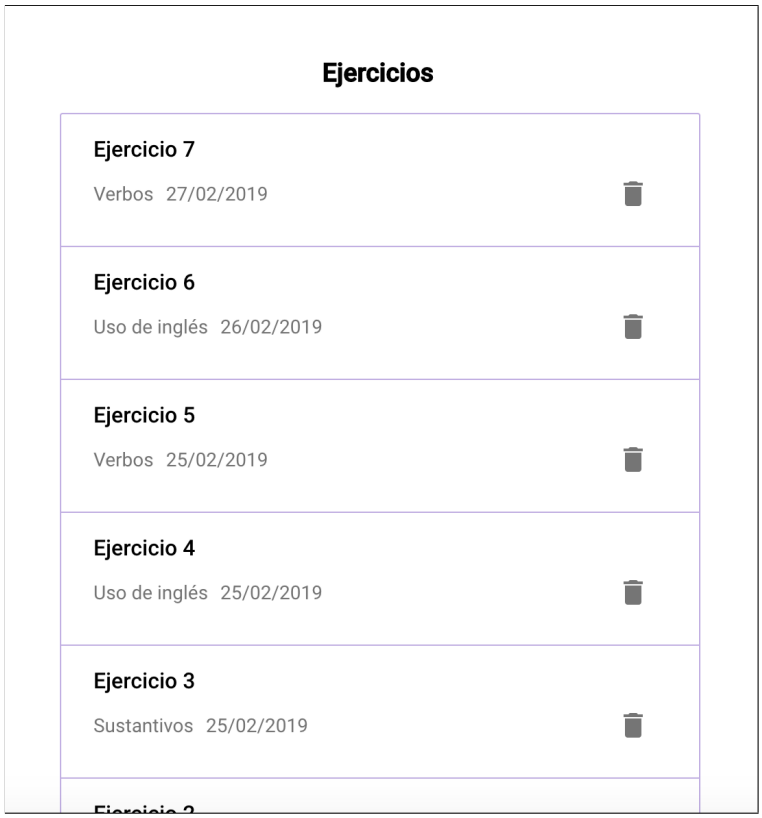

FIGURA 5.6: Lista de ejercicios almacenados.

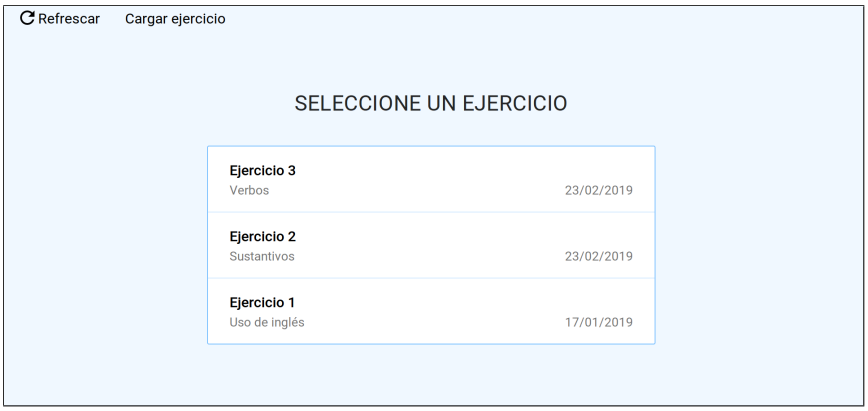

FIGURA 5.7: Inicio de la aplicación del alumno.

Con conexión

Si la computadora del alumno cuenta con conexión a internet, se despliega un menú que muestra los ejercicios que hay en la base de datos del servidor. También se da la opción de cargar un ejercicio localmente (ver figura 5.7). Además, la aplicación de escritorio guarda en una base de datos local los ejercicios descargados.

Sin conexión

Si no se cuenta con conexión a internet, primero desde la aplicación se busca en la base de datos local si hay ejercicios previamente descargados, en caso afirmativo se despliegan de la misma manera que cuando hay conexión a internet, en caso contrario se aclara que no se encontraron ejercicios y se da la opción de cargar un ejercicio desde un archivo local.

Para cargar el ejercicio desde un archivo, se utiliza el botón de "Cargar ejercicio" que permite seleccionar entre sus archivos el que contiene el ejercicio.

Pantalla de ejercicios

Una vez cargado el ejercicio, ya sea por medio del servidor o por el archivo local, la aplicación lee los metadatos que especifican el tipo de ejercicio y se muestra la pantalla con la actividad correspondiente pronta para ser realizada. Como el ejercicio ya contiene las respuestas cargadas, el alumno puede realizar el ejercicio aunque no tenga conexión a internet, y, una vez que finalice, es capaz de verificar que sus respuestas sean correctas o ver en qué partes se equivocó.

La aplicación tiene tres estilos distintos para completar los ejercicios. Los ejercicios de conjugación de verbos y uso del lenguaje muestran una interfaz donde sobre cada opción se despliega un menú con cuatro posibles soluciones como muestra la figura 5.8.

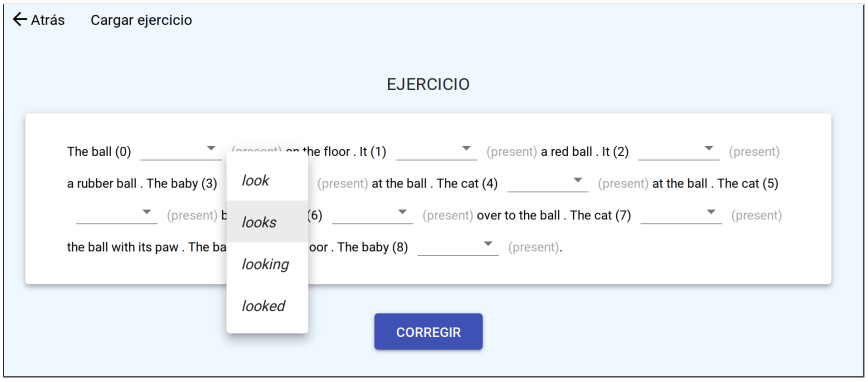

FIGURA 5.8: Selector de opciones en el ejercicio de uso del lenguaje.

Una vez que el alumno terminó la tarea puede ver las respuestas presionando el botón "Corregir". Esto lo puede hacer incluso si no completó todas las opciones. El ejercicio corregido se puede ver en la figura 5.9. En esta imagen se aprecia el ejercicio conjugación de verbos resuelto, donde el color verde indica que la opción seleccionada es la correcta y el color rojo marca lo contrario.

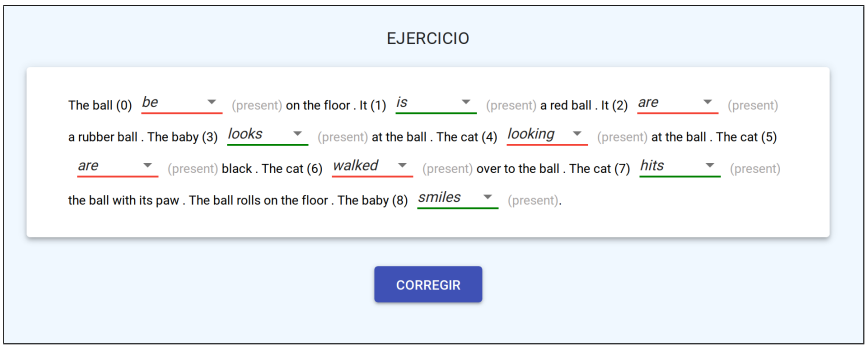

FIGURA 5.9: Ejercicio de uso del lenguaje corregido.

Por otro lado, el ejercicio de definiciones muestra primero todas las palabras posibles que hay para completar y debajo las definiciones con las espacios para completar. El alumno debe rellenar los espacios en blanco utilizando las opciones de arriba. Esta decisión fue intencional, para que de esta forma el alumno practique escribir las palabras.

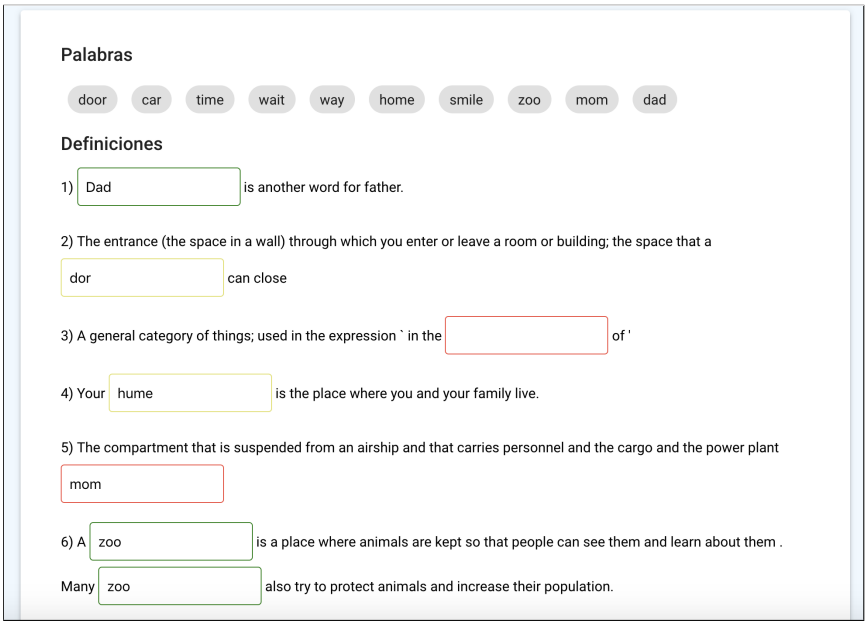

FIGURA 5.10: Ejercicio de sustantivos con respuestas corregidas.

En la imagen 5.10 vemos el ejercicio de sustantivos una vez que fue corregido. El color verde indica que la palabra es válida, el rojo que es incorrecta y el color amarillo le avisa al alumno que estuvo cerca, pero la palabra que escribió tiene una falta de ortografía.

Por último, el ejercicio de hipónimos presenta una interfaz más interactiva donde se debe arrastrar las palabras hacia su categoría correcta. Creemos que de esta forma va a ser más motivante y divertido para los alumnos realizar este tipo de ejercicio. Al corregir el ejercicio las palabras que estén en una categoría incorrecta serán marcadas con rojo y las correctas en color verde nuevamente (Ver 5.11).

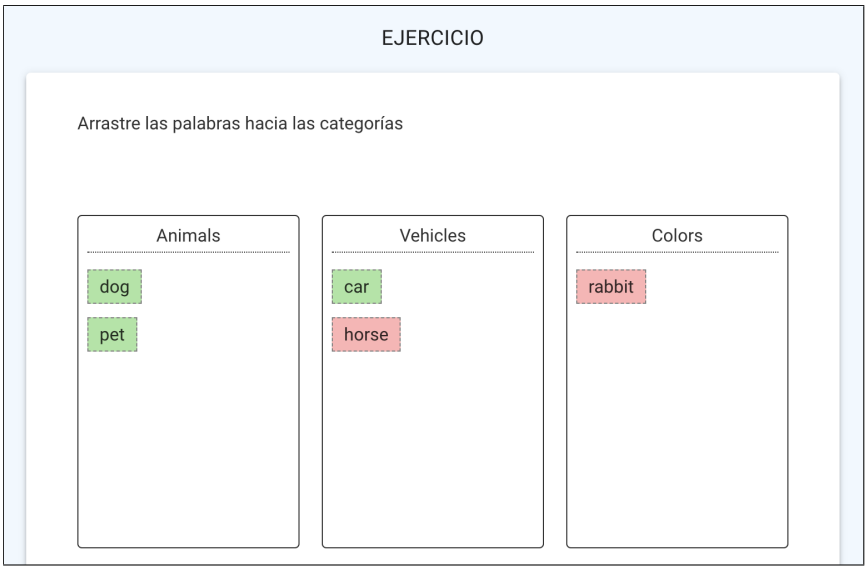

FIGURA 5.11: Ejercicio de hipónimos desde el lado del estudiante.

5.2. Arquitectura

La arquitectura se diseñó para contemplar todos los requisitos relevados (ver sección 3.1) y a su vez para estar preparada para que en un futuro sea utilizada por muchos usuarios en el contexto que se espera (docentes y alumnos en varias escuelas).

A alto nivel, la arquitectura se compone de tres elementos principales: el servidor, los docentes y los estudiantes (figura 5.12). Tanto el docente como el estudiante se comunican con el servidor para obtener y procesar los diferentes recursos que la plataforma les ofrece.

La decisión más importante sobre la arquitectura que hubo que tomar fue dónde debía ir el procesamiento de los ejercicios. Las dos opciones que nos planteamos que podíamos llevar a cabo fueron hacer que la aplicación de los docentes contenga este procesamiento o tener un servidor propio que lo haga.

La primera opción (procesamiento por parte de la aplicación de docentes) tiene como principal ventaja que los docentes no requieren de conexión para generar los ejercicios pero, por otro lado presenta a su vez varias desventajas que se listan a continuación.

- Limitantes en capacidad de memoria y espacio

  Nuestra plataforma demanda una capacidad de procesamiento considerable. Por lo que los docentes deberían contar con computadoras lo suficientemente potentes como para poder correrlo. Como uno de los requisitos es que sea compatible con todas las computadoras brindadas

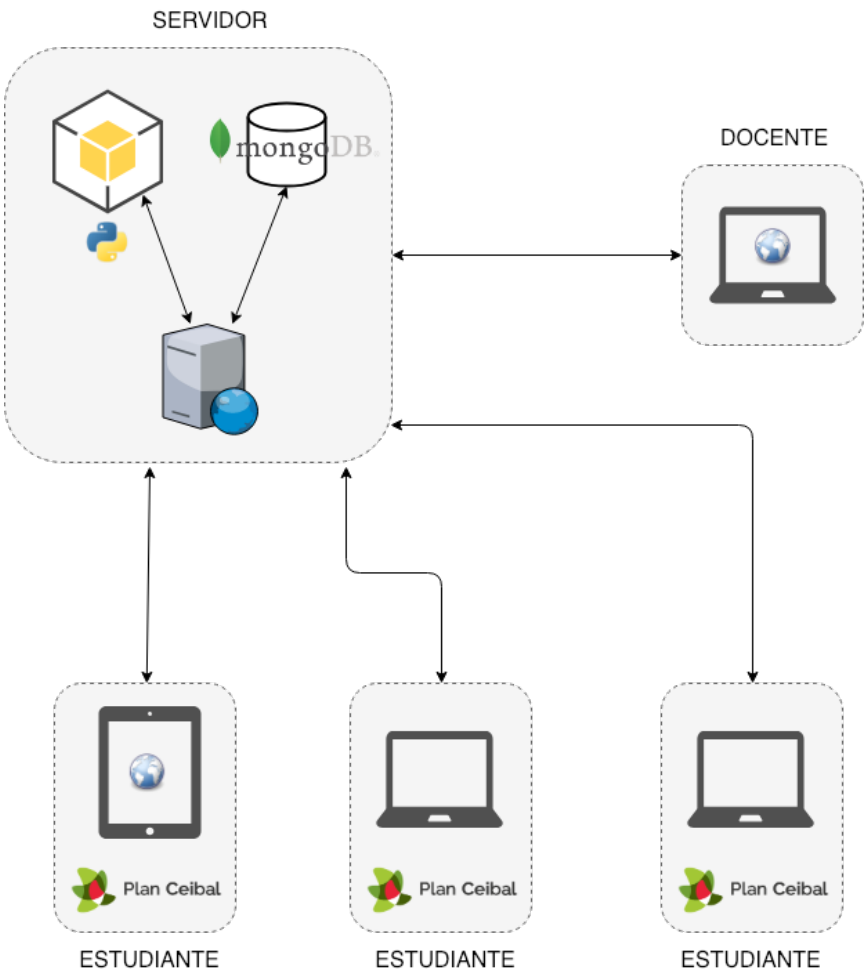

FIGURA 5.12: Arquitectura de la aplicación

por el Plan Ceibal [36] esta fue la más importante de las desventajas que se presentaron.

- Distribución de la aplicación

  La aplicación del docente tiene que poder ser instalable en las computadoras de los maestros y además ellos deberán contar con suficiente espacio en disco. Esto dificulta la distribución, ya que tiene que ser por un medio físico (pendrive, CD, etc.) o por almacenamiento en la nube, y la aplicación en sí es muy pesada. Si no es necesario el procesamiento en la aplicación del docente entonces se puede hacer una aplicación web lo cual simplifica mucho la distribución.

- Material compartido

  Como la base de datos es local a cada docente, no es posible compartir material generado con otros docentes como los ejercicios y textos. La única ventaja que esto presentaría es que cada docente podría modificar y cambiar las definiciones o el vocabulario a su gusto sin alterar el contenido del resto de los docentes.

- Distribución de ejercicios

  Si los ejercicios se encuentran solamente en la computadora del docente, la única forma posible para pasarlos es a través de un medio físico hacia la computadora de los estudiantes. En cambio si los ejercicios estuviesen almacenados en un servidor existiría la opción de acceder por medio de internet, en caso de que halla conexión.

Frente a la gran cantidad de desventajas con las que cuenta la primera alternativa decidimos implementar la segunda, que consiste en tener el procesamiento y la base de datos en un servidor fuera de la aplicación del docente y que sea accedida a través de internet. Además, la aplicación del estudiante con conexión también puede acceder a la base de datos. Con un servidor es posible asegurar que se tengan las condiciones necesarias para que todo el procesamiento funcione correctamente.

En cuanto a tecnologías del lado del servidor, el módulo de procesamiento lo implementamos con Python 2 junto con una base de datos MongoDB no relacional. La justificación a esta elección se debe a que Python tiene muchas librerías ya implementadas para el PLN.

La aplicación del docente es una aplicación web que puede ser accedida desde cualquier computadora con una URL pública y utiliza los servicios que brinda el servidor para sus funcionalidades.

Para los estudiantes desarrollamos una aplicación de escritorio para los sistemas operativos Linux y Windows compatibles con las computadoras del Plan Ceibal, y también una aplicación web disponible con una URL pública. La ventaja de la aplicación de escritorio sobre la web es que puede funcionar sin conexión, pero la aplicación web permite ser accesible desde otros dispositivos como tablets o celulares, y no necesita ser instalada. La aplicación en todas las plataformas consume las funcionalidades del servidor siempre que exista conexión a internet.

5.2.1. Arquitectura sin conexión

Uno de los requisitos fue que el estudiante pueda utilizar la plataforma sin conexión (ver 3.1). Para lograr este requisito diseñamos una variante de la arquitectura donde los estudiantes pueden cargar ejercicios que descargaron previamente por internet o mediante un archivo distribuido por el docente (ver figura 5.13).

Si el docente tiene acceso a internet puede generar nuevos ejercicios, y en caso contrario puede generarlos previamente y descargarlos en un archivo para distribuirlo con los estudiantes.

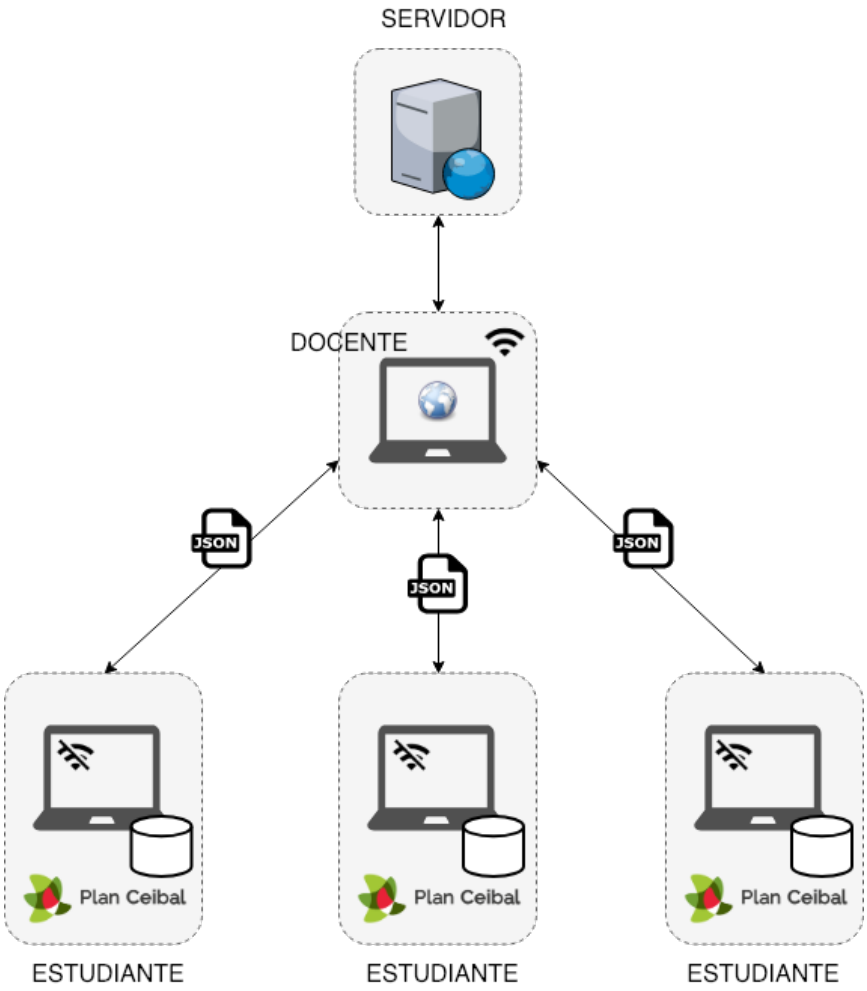

FIGURA 5.13: Arquitectura sin conexión de la aplicación

La aplicación de estudiantes cuenta con una base de datos local que guarda todos los ejercicios que se descargaron desde la última conexión a internet. Y a su vez permite cargar los ejercicios que el docente le brinda por algún medio físico dentro del salón de clases.

5.3. Distribución de la plataforma

Dadas las características de la arquitectura y el contexto donde se utilizaría la plataforma, no fue sencillo conseguir los recursos informáticos para distribuirla.

En etapas de desarrollo el servidor corría local en nuestras computadoras personales, pero para distribuirla a los maestros tuvimos que conseguir un servidor para que sea accesible desde cualquier lado. Como la facultad no nos pudo proveer de dicho servidor y era necesario para que los maestros puedan probar la aplicación decidimos utilizar los servicios de Amazon Web Services

(AWS), en particular Amazon Elastic Compute Cloud (EC2) que proporciona instancias de servidores virtuales.

Para la distribución de las aplicaciones web, tanto la del docente como la del estudiante, utilizamos otro servicio que provee Amazon Web Services (AWS) para el almacenamiento de sitios estáticos. Como la aplicación es accesible con una URL fue fácil que ambos usuarios puedan utilizarla.

La aplicación de escritorio para estudiantes cuenta con un instalador para los sistemas operativos Linux y Windows. Gracias a que el Plan Ceibal nos brindó dos computadoras como las que utilizan los alumnos, pudimos instalar y probar la aplicación en las mismas. El instalable tiene un peso de 39.5MB para Linux y 43MB para Windows, no tuvimos la oportunidad de poder distribuir el instalable con los estudiantes, pero afortunadamente los maestros pudieron probarlo en las distintas actividades que realizamos. Por eso decidimos distribuirlo a los estudiante utilizando la aplicación web que cuenta con las mismas funcionalidades con excepción de que la de escritorio puede funcionar sin conexión.

6. Actividades de vinculación

Como ya mencionamos anteriormente, nuestro proyecto de grado tuvo la particularidad de realizarse en el marco de una colaboración entre ANEP - UdelaR. La idea de este capítulo es documentar las diferentes actividades en las que participamos junto con nuestros tutores y el contacto que tuvimos con lo funcionarios de ANEP y los maestros de las escuelas.

6.1. Reuniones con ANEP

A principios del año 2018, antes de empezar la elaboración del proyecto, tuvimos una serie de reuniones con funcionarios del departamento de Políticas Lingüísticas de ANEP. Dichas reuniones tuvieron lugar en la sede de Políticas Lingüísticas, ubicada en la calle 18 de julio 841. Allí conocimos a Aldo Rodríguez, coordinador de Políticas Lingüísticas del Consejo Directivo Central, Codicen de ANEP, y a Valentina Dubini, coordinadora de enseñanza de inglés de escuelas rurales. En estas reuniones nos informaron en líneas generales sobre la realidad de las escuelas rurales. También nos comunicaron qué esperaban ellos del proyecto, discutimos algunos lineamientos y requisitos para nuestra plataforma y recibimos documentos e información acerca del nivel de inglés con que iban a tener que ser implementados los ejercicios. Además nos brindaron el contacto de dos escuelas rurales, con el objetivo de dar participación a dos maestros durante el desarrollo del proyecto y para que podamos recibir retroalimentación. Una de las escuelas está en el departamento de Lavalleja, es la escuela Rural Nº 25 de "Paso de los troncos", y la otra es la escuela Rural Nº 16 de "La Hilera", ubicada en el departamento de Tacuarembó. Mediante llamadas por Skype pudimos presentarnos frente a los maestros y conocer su opinión respecto al tema, además de coordinar futuras visitas a las escuelas.

Como el proyecto contaba con una contraparte involucrada e interesada y una realidad social referida a la educación de nuestro país, nos motivó a empezar a trabajar y mantener el contacto mediante estas reuniones y por correo electrónico con los funcionarios de ANEP. Otro factor que ayudó a dar el puntapié inicial para comenzar con el desarrollo fueron los materiales

que nos dieron, que nos sirvieron como base para buscar y elegir los ejercicios que después íbamos a implementar.

Para una de las oportunidades implementamos una prueba de concepto con el ejercicio de conjugación de verbos pero, lamentablemente, no pudimos mostrárselo a la maestra debido a problemas con la conexión de internet por lo que solo pudimos hablar por el altavoz de una llamada telefónica donde nos dio su opinión sobre el ejercicio. Sin embargo, pudimos mostrar el prototipo que teníamos a Valentina y a Colleen, una maestra de Estados Unidos de intercambio que se encontraba de visita por el país y estaba dando clases a algunos grupos en escuelas del interior.

El equipo de ANEP se mostró en todo momento muy flexible ante el tipo de trabajo a realizar y dispuestos a responder consultas, además de mostrarse verdaderamente interesados con el proyecto. Lo mismo para los maestros asignados. A mediados del 2018 recibimos por parte del Plan Ceibal dos laptops ceibalitas y una tablet para poder probar la aplicación.

6.2. Visita a Lavalleja

El día veinte de agosto de 2018, acompañados por nuestros tutores, Valentina Dubini, y una funcionaria del área de investigación y extensión de la Facultad de Ingeniería, visitamos finalmente la escuela ubicada en el departamneto de Lavalleja que nos había sido asignada desde el inicio. Una vez allí nos recibió Leticia Arbelo, la maestra que nos relató de primera mano cómo es la realidad en las escuelas rurales. Leticia es maestra directora, es decir, se encarga de impartir conocimientos de todas las áreas de estudio a sus alumnos, y, además, es la responsable de administrar la escuela, organizar el desayuno y el almuerzo para los niños, servirles la comida y mantener el establecimiento en buen estado. Es la única maestra en la escuela y a pesar de que probablemente haya algún otro empleado que la ayude, sus tareas son muy agotadoras para una sola persona.

Su clase la integran quince niños, desde los cinco a los doce años de edad. Al tener diferentes edades, cada niño está siguiendo años de curso distintos aunque se encuentren en el mismo lugar físico, por lo que la maestra tiene que preparar las lecciones para cada curso. Además de que ella tiene en cuenta las capacidades y las velocidades de aprendizaje de cada uno de los alumnos. La cantidad de estudiantes no alcanza para que asignen a otro maestro en la escuela. Ella vive en la escuela pero los niños hacen turno de tiempo completo y vuelven en la tarde a sus casas. Nos contó lo sacrificado que era para algunos niños tener que ir a la escuela desde tan lejos y de lo importante que es poder asistir.

Al estar sola no podía dejar el salón ausente por mucho rato por lo que no pudimos estar con ella todo el tiempo que hubiéramos querido. De todas

formas pudimos hablar con ella y mostrarle la aplicación, preguntarle si le serviría y nos dio su número para contactarla en el futuro. El ejercicio de conjugación de verbos le pareció un poco difícil para el nivel de sus estudiantes. Pero mencionó que varios de sus alumnos aprenden rápido y aprovechan para enseñar a los demás a través del ejemplo por lo que no sería ningún problema. Leticia también nos dijo que aunque el ejercicio parezca un tanto complicado, sus alumnos deberían hacerlo ya que los prepara para los nuevos desafíos que vendrán. Por otro lado, tanto el ejercicio de sustantivos como el de uso del lenguaje le gustaron y le parecieron adecuados para su clase. Nos propuso que en algún ejercicio les hagamos a los estudiantes reescribir las palabras para trabajar en la mejora de la ortografía, en lugar de solo tener que seleccionar la palabra correcta. Nos fuimos con la impresión de que Leticia quedó entusiasmada con la aplicación.

Valentina le entregó algunas tarjetas con vocabulario para poder usarlas en sus clases. Esta acción de llevarle materiales la realiza cada tanto que puede, que no es muy seguido, lo que reafirma la necesidad de contar con alguna manera de tener nuevos materiales para ejercicios.

FIGURA 6.1: Escuela rural Nº 25, Paso de los troncos

En la imagen 6.1 podemos apreciar el frente de la escuela. Con respecto a la conectividad a internet no tuvimos ningún problema a la hora de generar los ejercicios. Cada niño tenía su propia ceibalita, aunque había algunas que estaban en el taller para arreglar. La maestra también tenía la suya. Algo que nos hubiera gustado hacer y no tuvimos la oportunidad fue que los alumnos interactúen con la aplicación.

6.3. Foro de lenguas

Otra de las actividades que llevamos a cabo durante el proyecto fue asistir al 11º Foro de lenguas que se realizó los días 12 y 13 de octubre del 2018. El evento consistió en una serie de charlas y talleres sobre enseñanza de lenguas orientada a docentes de todos los ciclos. Al evento concurrieron diferentes ponentes, estudiantes de formación docente e investigadores en el tema. Además participaron autoridades importantes de la educación liderados por gente del Departamento de Políticas Lingüísticas de ANEP.

Se tocaron temáticas referidas a la enseñanza de distintas lenguas y también se desarrollaron actividades organizadas por estudiantes. El evento se encuentra dentro de ciertos lineamientos estratégicos que apuntan a la profesionalización y actualización de los docentes en el campo de los idiomas. Además apunta a compartir experiencias, informes e investigaciones, es decir tiene como objetivo convertirse en un punto de encuentro en el que se compartan y difundan las investigaciones y experiencias educativas a través de espacios de reflexión y debate. Dada su magnitud, el Foro de Lenguas se ha posicionado como un encuentro de referencia en la región, donde asisten invitados del extranjero.

En el marco del evento tuvimos la oportunidad de presentar nuestro proyecto en la sala de conferencias del Instituto de Perfeccionamiento y Estudios Superiores (IPES). Allí, junto a nuestros tutores, les presentamos a los maestros el trabajo que veníamos realizando y las ideas que teníamos para cuando el proyecto llegara a su fin. La foto 6.2 muestra uno de los momentos de la presentación. Al final de la charla realizamos una demostración en vivo con los ejercicios que teníamos implementados hasta el momento y le dimos la oportunidad a los presentes de probarla.

Los profesores que asistieron se mostraron muy motivados, con ganas de probar e implementar el uso de la aplicación. En la foto 6.3 se aprecia a los docentes escuchando durante la demostración. Muchos se interesaron tanto que dijeron que les serviría bastante una aplicación de este tipo para el día a día con sus alumnos, y aunque no todos eran maestros rurales, dijeron que lo usarían con sus estudiantes de secundaria. Algunos de ellos nos pasaron sus contactos para que, cuando la aplicación se encuentre lista para ser distribuida a las escuelas, puedan usarla ellos también.

FIGURA 6.2: El equipo durante la charla en el foro de lenguas.

FIGURA 6.3: El equipo y los maestros durante la demostración en el foro de lenguas.

6.4. Encuesta a docentes

Para ver cómo resultaban las interacciones entre la plataforma y los usuarios, les pedimos a distintos docentes que la probaran y respondieran ciertas preguntas acerca de ella. Dentro de esta sección detallamos las preguntas que hicimos, y explicamos de forma resumida la retroalimentación que recibimos por parte de los entrevistados.

Entre los entrevistados contamos con maestros que trabajan hace más de un año, estudiantes de magisterio y profesores de inglés. Uno de ellos es docente en una escuela rural. En total se obtuvieron respuestas de siete docentes.

Preguntas de la encuesta

Seguidamente se listan las preguntas que realizamos en la encuesta. El propósito de las mismas era conseguir comentarios y sugerencias por parte de los docentes.

1. ¿Cuál es el nombre de la escuela o institución?
2. ¿Te pareció útil la propuesta?
3. ¿Usarías el ejercicio de conjugación de verbos con tus alumnos?
4. ¿Usarías el ejercicio de sustantivos con tus alumnos?
5. ¿Usarías el ejercicio de uso del lenguaje con tus alumnos?
6. ¿Usarías el ejercicio de hipónimos con tus alumnos?
7. ¿Por último, que le cambiarías o le agregarías a cualquiera de las dos aplicaciones?

Todas las respuestas de las preguntas junto con comentarios recibidos se encuentran en el anexo C.

Comentarios recibidos

A todos los docentes encuestados les pareció útil la propuesta, aunque los motivos fueron variados. Algunos plantearon que simplifica la tarea docente, destacando la rapidez con que genera ejercicios. Otros plantearon que sirve para ocupar espacios ociosos y para manejar mejor las diferencias entre niveles de estudiantes. Otro aspecto positivo que remarcaron es lo fácil

de usar que es la aplicación. Entre las recomendaciones generales, también nos dijeron que incorporar algún tipo de material audiovisual serviría para poder trabajar con niños más pequeños. Sin embargo, este aspecto no estaba contemplado en el alcance de nuestro proyecto.

Recibimos buena retroalimentación sobre el ejercicio de sustantivos y el de uso del lenguaje. Sobre el de sustantivos opinaron que se debe acompañar con imágenes para esclarecer conceptos, y el de uso del lenguaje debería ser revisado por los maestros para controlar que la dificultad no sea excesiva. Sobre el ejercicio de conjugación de verbos, nos propusieron eliminar la sugerencia de tiempo verbal. Esto lo discutimos pero, por las razones explicadas en la sección de ejercicios, consideramos que era mejor dejarlo como estaba. Por último, en el ejercicio de hipónimos algunos docentes nos manifestaron que no se encontraba ninguna palabra para las categorías o que tenía pocas palabras como para que sea útil para los estudiantes. En la sección del ejercicio de hipónimos indicamos este problema y planteamos posibles soluciones.

La figura 6.4 muestra un heatmap con los resultados acerca de si usarían o no los ejercicios. Se puede apreciar que la gran mayoría usaría los cuatro ejercicios, donde el ejercicio de uso del lenguaje es el que cuenta con la aprobación de todos los docentes, lo cual es alentador.

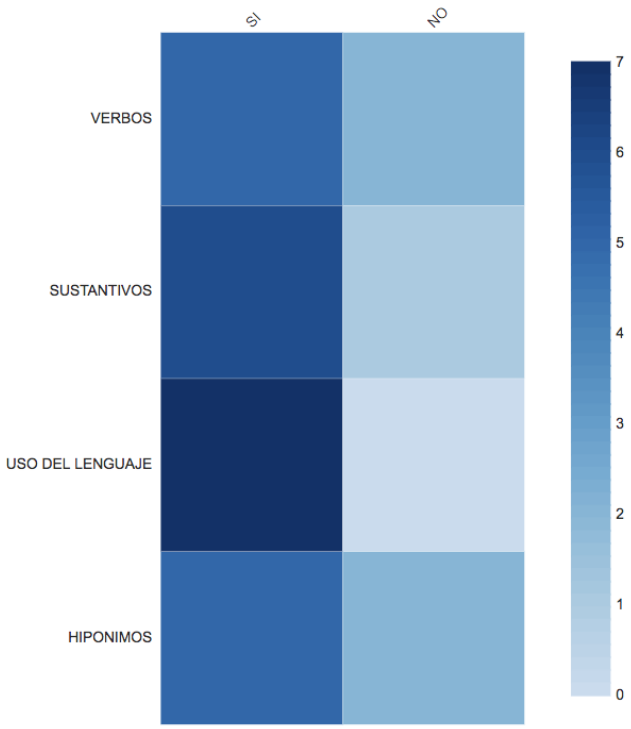

FIGURA 6.4: Heatmap que muestra los resultados de la encuesta referidos a los ejercicios.

7. Conclusiones

7.1. Conclusiones

Este trabajo aborda la investigación de técnicas de PLN junto con procedimientos asistidos por computadora para la construcción de una herramienta de generación automática de ejercicios. En primer lugar podemos afirmar que construimos una plataforma que colabora con la misión de alcanzar los objetivos propuestos por ANEP respecto a la enseñanza de inglés. Aunque aún no fue posible implantar y ver los resultados de la misma sobre los estudiantes.

A lo largo del proyecto nos fuimos dando cuenta de la necesidad de generar buenos recursos. Los corpus son difíciles de conseguir y muy valiosos para aumentar la calidad y eficiencia de las aplicaciones. En nuestro trabajo para lograr superar este obstáculo tuvimos que ajustar un corpus muy grande y con demasiado vocabulario como el de Wikipedia para que sea compatible con un rango de niveles de inglés específico. Pero sin lugar a dudas, se podrían haber obtenido los mismos o mejores resultados desde el comienzo si hubiéramos tenido un corpus de textos apropiado para trabajar en un nivel inicial de inglés.

Otra importante lección aprendida es que constantemente hay que hacer un compromiso que comprenda por un lado la calidad de los recursos y por otro el tamaño y la velocidad de procesamiento. En varias ocasiones tuvimos que realizar cambios significativos para alcanzar este balance. Redujimos el volumen del modelo de lenguaje y de los word embeddings. También agregamos la posibilidad de alternar entre el uso de un POS tagger u otro ya que según el servidor que se tenga las velocidades de respuesta pueden ser o no aceptables para la dinámica de un salón de clases.

Por los métodos de PLN que utilizamos para construir la plataforma, no aspiramos a que sea infalible, o que genere ejercicios sin necesidad de supervisión por parte de un humano. Es por esto que agregamos la funcionalidad de que los maestros puedan editar los ejercicios generados.

En la experimentación con la plataforma ajustamos algunos parámetros

para que la generación de ejercicios funcione de forma óptima para los niveles de inglés especificados, el modelo de lenguaje y el modelo word embeddings. Sin embargo, sabemos que el valor de estos parámetros debe ser modificado si alguno de estos tres recursos cambia con el fin de mantener la calidad de los ejercicios generados. Teniendo esto en cuenta permitimos fácilmente modificar estos parámetros.

La retroalimentación que recibimos, tanto en las entrevistas como a lo largo de todas las actividades compartidas con los docentes, fue muy positiva, y logramos el objetivo de contar con una herramienta intuitiva, que es sencilla para entender como usarla y que implica pocos pasos. Los maestros se mostraron entusiasmados a la hora de probar la aplicación. El uso de esta aplicación ahorra claramente en tiempo de confección de actividades y puede llegar a brindar un variado repertorio de las mismas para que los alumnos realicen, además de poder compartir con otros docentes los ejercicios. Podemos afirmar entonces que hay una gran necesidad por este tipo de tareas aplicada a la enseñanza de lenguas.

Sobre la aplicación de los alumnos consideramos que no tuvimos suficiente retroalimentación, ya que sólo la probaron docentes y no estudiantes que se encuentren aprendiendo el idioma.

Por nuestra parte, consideramos haber cumplido con éxito los requisitos que se plantearon y logrado satisfacer a las partes interesadas en este proyecto, sin embargo, sabemos que el objetivo real detrás de este trabajo es lograr mejorar el nivel de inglés de los alumnos en escuelas rurales y para eso aún hay mucho camino que recorrer. Recibimos aceptación por parte de los maestros, por lo que el próximo paso será lograr que se utilice la plataforma en las escuelas rurales. Creemos que este proyecto abre las puertas para continuar con la investigación e implementación de herramientas que usan el PLN en la enseñanza de idiomas.

7.2. Trabajos a futuro

Como posibles mejoras o líneas de trabajo a continuar para aumentar la calidad de los ejercicios y mejorar la distribución de la aplicación queremos destacar los siguientes puntos.

Procesamiento de los textos

Si uno de los futuros objetivos es extender la plataforma hacia otros niveles de inglés, entonces un trabajo a futuro interesante sería implementar un simplificador de textos para los niveles más bajos de inglés. La simplificación de texto se puede ver como una tarea de traducción donde el lenguaje origen

es el idioma con todo el vocabulario y el lenguaje destino es el idioma con un vocabulario reducido y sintaxis más simple.

Por otro lado, algo que investigamos fue poder detectar la dificultad del nivel de un texto, pero no vimos necesario agregarlo para este proyecto ya que los docentes siempre trabajan dentro un mismo rango de niveles de inglés. Si en un futuro hay distintos niveles y más variados, sería muy útil poder detectar el nivel del texto ingresado para luego por ejemplo asociarlo con las categorías de Cambridge y alertar al docente en el caso de que fuera a utilizarlo con otro nivel diferente de estudiantes.

Por último, otra posible mejora que nos gustaría realizar es la que permita identificar estructuras o elementos gramaticales dentro del texto antes de que se pueda elegir qué tipo de ejercicio hacer. De esta forma, solo se sugerirían ejercicios que se puedan generar a partir de un texto en particular. Por ejemplo si al cargar un texto, se encuentra que éste no contiene ninguna palabra que pertenezca a las categorías de concepto preestablecidas entonces la opción de generar el ejercicio de hipónimos no se muestra al docente.

Corpus

Una de las dificultades que tuvimos a la hora de generar los ejercicios fue encontrar corpus adecuados para los niveles de inglés requeridos.

Un corpus de textos orientado al vocabulario de los niños podría mejorar la efectividad de los recursos que tenemos generados. Como ya mencionamos en secciones anteriores (ver 6, 3.3.7), el modelo word embeddings tiene vocabulario complicado, y la lista de frecuencias tiene palabras que comúnmente no son muy utilizadas hoy en día.

Además, si contáramos con un corpus de errores, podríamos detectar las dificultades comunes que presentan los alumnos y generar distractores incluso más apropiados aún para que ataquen y fortalezcan esas dificultades.

Sistema de usuarios

Además de la funcionalidad de importar y exportar ejercicios, pensamos que sería de utilidad que la aplicación permita que los docentes tengan cuentas propias de usuarios para poder visualizar los ejercicios creados por cada uno en un solo lugar sin interferir con los generados por otros docentes. Sumado a eso se podría agregar que los docentes editen el diccionario o el vocabulario a su gusto sin modificar o hacer una parte de los recursos por sesión y otra parte que sea compartida entre todos los usuarios.
Por otro lado si los estudiantes también pudieran contar con cuentas de usuarios se podría guardar los ejercicios resueltos, para ir midiendo el progreso

que han hecho de forma que el maestro les pueda ir asignando cada vez tareas más complejas.
Al no poder implementar esto preferimos concentrarnos de lleno en la funcionalidad de trabajar sin internet por parte de los alumnos.

Agregar nuevos tipos de ejercicios

Desde un principio, a partir de la investigación inicial realizada, contamos con una lista de ejercicios posibles para generar pero dado el alcance del proyecto, sólo se pudieron llevar a cabo cuatro de ellos.

Algunos ejemplos de nuevos ejercicios podrían ser:

- Encontrar palabras que tengan el mismo significado o sean sinónimas. Esto podría ser que a partir de dos listas de palabras, unir cada palabra de la primera lista con su sinónimo que se encuentra en la segunda lista.
- Dado un grupo de palabras donde todas menos una tienen algo en común, como por ejemplo la categoría conceptual, hacer que el alumno encuentre la que desentona en el conjunto.
- Ejercicios de comprensión lectora. Este tipo de ejercicio en particular creemos que es un verdadero desafío y podría ser interesante para los alumnos. Por ejemplo se podría implementar un ejercicio que hiciera preguntas acerca de información que se encuentra explícita en el texto.

Servidor

Nos vimos limitados por la capacidad del servidor, tanto en lo que refiere a los aspectos de memoria como de espacio para almacenamiento. Nos encontramos en la situación de que cada vez que hacíamos una mejora en generación de los ejercicios perdíamos en capacidad y en memoria, por lo que había que aumentar uno de los dos aspectos, sino los dos. Uno de los ejemplos más claros fue cuando utilizamos un servidor externo para poder dejar pública la aplicación de los docentes para que ellos la probaran. Como era un servidor bastante chico con 4 GB de RAM y 30GB de espacios de disco tuvimos que sacrificar algunas mejoras que habíamos realizado como agregar un parser más pesado o mejorar el modelo de word embedings para que el servidor pueda correr.

En particular para ese caso, tuvimos que cambiar el parser de Stanford por el de NLTK y reducir el tamaño del modelo de word embeddings que no arroja tan buenos resultados como uno de mayor de volumen.

Estamos seguros que de si en un futuro contáramos con un servidor sin limitantes de infraestructura se van a poder conseguir mejores resultados todavía, tanto con los ejercicios generados como con los tiempos de respuesta.

Además un servidor con estas características permitiría llevar a cabo una distribución de las aplicaciones más segura y confiable, sin temor a que el servidor no les funcione a los docentes o a los estudiantes.

Glosario

ANEP

(Administración Nacional de Educación Pública) Organismo del Estado uruguayo que se encarga de la planificación, gestión y administración del sistema educativo público en nivel inicial, primaria, media, técnica y formación en educación terciaria.

CALL

(Computer assisted language learning) Aprendizaje del lenguaje asistido por computadora, se centra en la búsqueda y estudio de los usos de las computadoras para ese aprendizaje.

código abierto

Código que permite la colaboración por parte de cualquier persona para su elaboración y mejora.

distractor

Es una respuesta incorrecta en un ejercicio de múltiple opción. Muchas veces están diseñados a partir de errores comunes o dificultades que el estudiante se pueda encontrar.

Flyers

Es la tercera y la más avanzada de las tres pruebas de la Universidad de Cambridge dirigidas a los alumnos de nivel escolar.

hipónimo

Término que incluye en su definición el significo de otro término más general además de sus propios atributos semánticos.

MCER

(Marco Común Europeo de Referencia para las lenguas) Es un estándar para medir el nivel de comprensión y expresión oral y escrita en una determinada lengua.

morfema

Unidad significativa más pequeña que conforma a las palabras.

Movers

Es la segunda de las tres pruebas de la Universidad de Cambridge dirigidas a los alumnos de nivel escolar.

NLTK

Es un conjunto de librerías y programas para el procesamiento de lenguaje natural para el lenguaje de programación Python.

parser

Herramienta que se encarga de realizar el proceso de parsing.

parsing

Proceso durante el cual luego del análisis morfológico el texto se puede convertir en otras estructuras de trabajo.

PLN

(Procesamiento de lenguaje natural) Rama de la Inteligencia Artificial que en pocas palabras se encarga de la comprensión y la generación del lenguaje.

POS

(Part of Speech) Categoría de palabras que comparten propiedades gramaticales como pueden ser, verbos, sustantivos, etc.

PPL

(Programa de políticas lingüísticas) Programa de ANEP que tiene como meta formar ciudadanos plurilingües.

Starters

Primera de las tres pruebas de la Universidad de Cambridge dirigidas a los alumnos de las escuela.

synset

Representación de un concepto dentro de la base de datos léxica WordNet.

taggear

Proceso durante el cual se lleva a cabo el etiquetado gramatical de una secuencia de palabras.

tagger

Recurso que se encarga de llevar a cabo el proceso de etiquetado de categorías gramaticales de una secuencia de palabras.

token

Secuencia de caracteres, por ejemplo las palabras.

tokenizar

Los textos, en general, entran en una aplicación como un solo string. Tokenizar es la acción de transformar ese string en una lista de palabras, para poder manipular el texto a nivel de palabra y no a nivel de carácter.

WordNet

Base de datos léxica del idioma inglés.

WSD

(Word Sense Disambiguation) Problema de identificar el significado correcto de una palabra dentro de un cierto contexto.

A. Procesamiento corpus

En este anexo se muestra el script construido para obtener el corpus de Wikipedia a partir de un archivo dump xml.

A.1. Script procesamiento corpus

En primer lugar es necesario contar con el dump de Wikipedia, que es posible descargarlo desde https://dumps.wikimedia.org/enwiki/latest/.

Luego, es necesario instalar el paquete de Gensim [47].

```
$ pip install gensim
```

A continuación se muestra el script:

```
import sys
from gensim.corpora import WikiCorpus

def make_corpus(in_f, out_f):

  """Convertir Wikipedia xml dump en un corpus de texto"""

  output = open(out_f, 'w')
  wiki = WikiCorpus(in_f, lemmatize=False, dictionary={})

  i = 0
  for text in wiki.get_texts():
    output.write(bytes(' '.join(text), 'utf-8').decode('utf-8') + '
        \n')
    i = i + 1
    if (i %10000 == 0):
      print('Processed ' + str(i) + ' articles')
  output.close()
  print('Processing complete!')


if __name__ == '__main__':
  in_f = sys.argv[1]
  out_f = sys.argv[2]
  make_corpus(in_f, out_f)
```

Para utilizarlo se ejecuta desde la línea de comandos ingresando el dump xml como parámetro.

```
$ python generar_corpus_wiki.py <wikipedia_dump> <texto-procesado>
```

B. Niveles de inglés

En esta sección se pretende informar al lector sobre los distintos niveles o categorías de grado de inglés en los que se pueden encontrar los estudiantes, los cuales nos basamos a la hora de desarrollar cada ejercicio. Es decir, hubo que tener en cuenta el vocabulario y la dificultad de los niveles para crear ejercicios que estuvieran acorde con los niveles esperados por parte de los alumnos y los maestros.

El Marco Común Europeo de Referencia para las lenguas (MCER) es el sistema encargado de definir y explicar los diferentes niveles de expresión y comprensión oral y escrita para lenguas como el inglés. Este estándar se utiliza principalmente en Europa, pero permite comparar la puntuación o nivel obtenidos en exámenes tales como Cambridge, IELTS, etc. El MCER comprende en total 6 niveles distintos, que se pueden apreciar en la imagen B.1 [49].

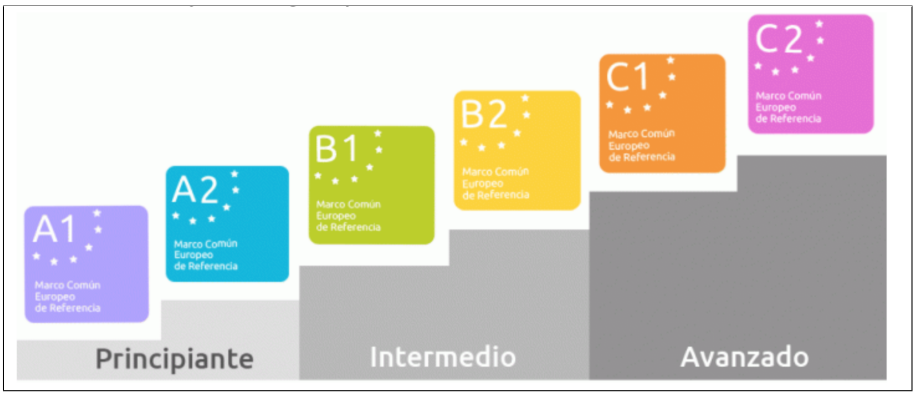

FIGURA B.1: Niveles de inglés dentro del MCER. Foto extraída del material brindado por ANEP.

El nivel A0 se corresponde con el de los principiantes. Allí se encuentran las personas que tienen nulo o muy poco conocimiento del idioma y que simplemente conocen algunas palabras sueltas del vocabulario.

El nivel A1 se corresponde con los usuarios básicos con el idioma, es decir, aquellos que tienen la capacidad de comunicarse en situaciones de la vida cotidiana con expresiones de uso frecuente y utilizando vocabulario básico y elemental. Las características que una persona en A1 debe cumplir son:

- Ser capaz de presentarse a sí mismo y a otros, pedir y dar información personal básica sobre su domicilio, sus pertenencias y las personas que conoce.
- Puede relacionarse de forma sencilla siempre y cuando su interlocutor procure hablar lento y con claridad y esté dispuesto a cooperar.
- Puede comprender y utilizar expresiones cotidianas de uso muy frecuente así como frases sencillas destinadas a satisfacer necesidades de tipo inmediato, como pedir un vaso de agua o ir al baño.

Nivel	Categoría	Descriptor
A0	IN PROGRESS	El estudiante está en proceso de aprendizaje de inglés y requiere más tiempo para iniciar sus progresos con el idioma.
A1-	BEGINNER LOW	El estudiante demuestra una comprensión inicial de vocabulario básico en inglés y de oraciones y estructuras gramaticales muy sencillas referidas a sí mismo, su familia y su entorno.
A1+	BEGINNER HIGH	El estudiante demuestra una comprensión consolidada de vocabulario básico en inglés y de oraciones y estructuras gramaticales sencillas referidas a sí mismo, su familia y su entorno.
A2-	ELEMENTARY LOW	El estudiante distingue y demuestra comprensión de diversas expresiones familiares en inglés y reconoce algunas estructuras de la gramática más frecuentes y esenciales referidas a sí mismo, sus actividades, sus gustos y su entorno.
A2+	ELEMENTARY HIGH	El estudiante distingue y demuestra comprensión consolidada de diversas expresiones familiares en inglés y reconoce las estructuras de la gramática más frecuentes y esenciales referidas a sí mismo, sus actividades, sus gustos y su entorno.

FIGURA B.2: Descriptores y niveles que se espera obtener en las pruebas de vocabulario, lectura y gramática. [50]

El último nivel que nos interesa es el A2, incluye además a las personas que son capaces de entender y comunicarse en situaciones simples, como por ejemplo presentarse a sí mismo y a su familia y relatar situaciones del pasado usando vocabulario básico. Dentro de las cualidades encontramos:

- Poder describir en términos simples algunos aspectos básicos de su pasado y su entorno así como cuestiones relacionadas con sus necesidades inmediatas.

- Entender frases y expresiones de uso frecuente relacionadas con áreas de experiencia que le son especialmente relevantes (información básica sobre sí mismo y su familia, compras, lugares de interés, ocupaciones, etc.).
- Es capaz de poder expresarse a la hora de realizar actividades simples y cotidianas que no requieran más que intercambios sencillos y directos de información sobre cuestiones que le son conocidas o habituales.

Nivel	Categoría	Descriptor
A0	IN PROGRESS	El estudiante está en proceso de aprendizaje de escritura en inglés y requiere más tiempo para progresar en su producción escrita.
A1-	BEGINNER LOW	El estudiante puede escribir palabras aisladas en inglés y está en proceso de comenzar a escribir oraciones sobre temáticas muy familiares.
A1+	BEGINNER HIGH	El estudiante puede escribir oraciones muy básicas en inglés sobre temáticas muy familiares, puede dar información muy básica sobre sí mismo y sobre otra persona.
A2-	ELEMENTARY LOW	El estudiante puede escribir oraciones en inglés sobre temáticas familiares, con algunos detalles, haciendo un uso adecuado de sintaxis y vocabulario.
A2+	ELEMENTARY HIGH	El estudiante puede escribir varias oraciones en inglés con diversos detalles sobre una variedad de temáticas familiares, haciendo un uso correcto de sintaxis y vocabulario. Puede escribir utilizando elementos de cohesión.

FIGURA B.3: Descriptores y niveles que se espera obtener en una prueba de escritura. [50]

La figura B.2 muestra las características que los estudiantes deben poseer para alcanzar cada uno de los niveles con respecto a la escritura, lectura, vocabulario y gramática de la lengua y la figura B.3 los aspectos respecto a la escritura.

La Universidad de Cambridge ofrece a los alumnos de educación primaria una serie de pruebas divertidas y motivadoras llamadas Young Learners English para el idioma inglés. Son tres diferentes pruebas basadas en actividades: Starters, Movers y Flyers. Esto les da a los estudiantes un camino claro

para mejorar en inglés desde principiantes. En la imagen B.4 podemos apreciar la correspondencia con los niveles de MCER y los de Cambridge, donde el nivel Starters se encuentra entre medio de A0 y A1 (como un pre A1).

Niveles MCER	Exámenes CAMBRIDGE	
		YLE Starters
A1 Acceso		YLE Movers
A2 Plataforma	KET	YLE Flyers
B1 Umbral	PET	
B2 Avanzado o usuario independiente	FCE	
C1 Dominio operativo eficaz	CAE	
C2 Maestría	CPE	

FIGURA B.4: Descriptores y niveles que se espera obtener en una prueba de escritura. Foto extraída del material brindado por ANEP.

Desde 2013, ANEP forma parte de los centros autorizados por la Universidad de Cambridge para emitir certificados a estudiantes de todos los subsistemas. Los niños de entre 9 a 11 años realizan exámenes Young Learners English desde nivel pre A1 al A2. Para terminar recordamos que el nivel A2 es el objetivo de egreso para educación primaria, pero que se espera se cumpla para la totalidad de los alumnos recién para el año 2030 según el estado actual de la enseñanza.

C. Entrevistas a los maestros

En esta sección publicamos las entrevistas realizadas a los docentes. También agregamos comentarios recibidos que no formaron parte de las preguntas, pero que nos resultaron interesantes.

Preguntas

1. ¿Cuál es el nombre de la escuela o institución?
2. ¿Te pareció útil la propuesta?
3. ¿Usarías el ejercicio de conjugación de verbos con tus alumnos?
4. ¿Usarías el ejercicio de sustantivos con tus alumnos?
5. ¿Usarías el ejercicio de uso del lenguaje con tus alumnos?
6. ¿Usarías el ejercicio de hipónimos con tus alumnos?
7. ¿Por último, que le cambiarías o le agregarías a cualquiera de las dos aplicaciones?

Respuestas

Micaela

1. Colegio y Liceo Latinoamericano.
2. Sí.
3. No.
4. Sí.
5. Sí.
6. No.

7. Quizás al ejercicio de conjugación de verbos le sacaría los paréntesis indicando el tiempo verbal. Sin embargo, al ser posible varias respuestas de esta forma, tal vez sería más útil sacar los verbos que hay que conjugar en cierto tiempo (pretérito perfecto por ejemplo) y dejar entre paréntesis su forma en infinitivo. Otra opción también podría ser sacar los verbos en dos tiempos y poner el verbo en infinitivo en paréntesis para que el alumno decida cuál es el correcto (por ej, pasado y presente en un mismo ejercicio).

Nicolasa

"Probé un texto típico de uso en clase de inglés y formé varias opciones de ejercicios con el mismo, y me sorprendió que lo hace muy bien y me viene muy bien para este año que voy a dar clases en un instituto de inglés. Veo la aplicación como una herramienta útil para ciertos ejercicios que suelen hacerse bastante seguidos y generar algo rápido que fácilmente podés modificar detalles si lo deseas."

1. Colegio y Liceo Latinoamericano.
2. Sí, muy útil y fácil de usar.
3. Sí, quizá realizaría modificaciones dependiendo del texto que use, pero genera los ejercicios adecuadamente.
4. Sí, con modificaciones también de ser necesarias.
5. Sí.
6. No.
7. Para mí lo bueno de la aplicación es que los ejercicios que genera son ejercicios típicos como en cualquier libro de inglés, y útiles. Acortan tiempo al docente, lo hace de manera más que deseable. No les cambiaría nada, pero quizá en un futuro se puedan agregar más tipos de ejercicios. Me gustó mucho la aplicación, no tuvo errores con el texto que usé.

Lucía

1. Escuela Nº 19, Grupo Escolar Felipe Sanguinetti.
2. La prospuesta me pareció muy interesante pero habría que adecuarla a los niveles de conocimiento general de los estudiantes y de sus edades.

3. Para este tipo de ejercicio depende del conocimiento de inglés que tengan los alumnos. Si bien ellos están acostumbrados a realizar pruebas online usando múltiple opción, las que requieren mucha lectura, se les complica cuando están en algunos grados recién aprendiendo a leer, como es el caso de mis alumnos (niños de 7 u 8 años).

4. Lamentablemente, por falta de tiempo no pude probarlo con mis alumnos pero por lo general, a nivel escolar se utilizan mucho las ilustraciones que ayudan al alumno y se lo hacen atractivo. El caso de los sustantivos sería un claro ejemplo de ello.

5. Usaría el ejercicio, pero previamente se debería realizar un diagnóstico del nivel de inglés que dominan los alumnos ya que algunos carecen de conocimiento de inglés y otros estudian en forma particular contando con una mayor ventaja.

6. Ídem al anterior.

7. En el caso de la aplicación "estudiantes" y a nivel escolar, incluso en los primeros años de Secundaria le aplicaría un estilo más visual más atractivo, destacando la diagramación, los colores, dibujos, etc. A nivel de primaria sobre todo en los primeros grados, los ejercicios de mucha lectura les resulta dificultosos.

Paula

"La aplicación resulta, de todas formas, apropiada, fácil de utilizar desde las dos partes, pero por su diseño y lo ya mencionado podría pensar que tendría mayor uso en ciclo básico y últimos grados escolares que en grados inferiores.

Supongo que la aplicación está diseñada para abrirse desde computadoras y sistemas Ceibal, desde el móvil la visualización del texto de sustantivos aparecía cortada impidiendo una lectura completa del texto (en la app de la maestra). "

1. Soy estudiante de 4to año de magisterio, cursando la práctica en la escuela nº 185.

2. Sí, la variedad de ejercicios aumenta la utilidad de la plataforma.

3. Todos los ejercicios pueden ser utilizados en el salón de clase, por sus características, presentación y formato considero que sería más exitosa en grupos superiores a tercer grado.

4. Ídem a la pregunta 3.

5. Ídem a la pregunta 3.

6. Ídem a la pregunta 3.

7. Les niñes reciben estímulos visuales incesantemente y por la diversidad de niveles y formas de aprender que se encuentran en un mismo salón de clase, considero que la aplicación podría tener la opción de agregar o contar por defecto con elementos visuales o auditivos que puedan tanto favorecer el aprendizaje cómo motivar al alumne a realizar la actividad. Así como podría tener una opción para graduar el nivel de dificultad del ejercicio (esto podría variar en la cantidad de espacios en blanco dentro de un mismo texto), con tres niveles dificultad la docente puede asegurarse de presentar una misma actividad accesible para todos les niñes de la clase.

Verónica

Verónica del Colegio y Liceo Bilingüe Ciudad Vieja no respondió la encuesta pero nos dio sus opiniones sobre la plataforma.

- La usaría pero que incluiría otro tipos de habilidades y no solo gramática.
- Es una buena herramienta para trabajar con la diversidad del alumnado, por ejemplo para aquellos alumnos que terminan antes el trabajo marcado en clase y aprovechar el tiempo ocioso.
- Es una actividad que los alumnos puede realizar en forma autónoma y resulta motivadora ya que el alumno va viendo los avances.
- Se puede utilizar como una tarea domiciliara.
- Puede ser utilizada cuando se da ausentismo docente o cuando el profesor suplente no tiene mismo nivel de inglés que el efectivo.

Gabriela

1. La escuela donde trabajo es la Nº 77 de Tapes Grande.
2. Me gustaron mucho las propuestas. Me parecieron amigables y muy accesibles.
3. Sí.
4. Sí.
5. Sí.
6. Sí.

7. Le agregaría si se pudiera alguna actividad con sonido, imagen o propuesta con video.

Ana

1. Colegio y Liceo Bilingüe Ciudad Vieja.
2. Sí.
3. Sí.
4. Sí.
5. Sí.
6. Sí.
7. Incorporar otras habilidades tales como imágenes o videos.

Bibliografía

[1] *Entrevista a Aldo Rodríguez sobre la enseñanza en escuelas rurales*. URL: https://educacion.ladiaria.com.uy/articulo/2018/9/autoridades-preparan-un-proyecto-para-que-el-ingles-llegue-a-todas-las-escuelas-rurales/.

[2] *Sitio oficial de Ceibal en Inglés*. URL: https://ingles.ceibal.edu.uy/.

[3] *Plataforma educativa CREA*. URL: https://blogs.ceibal.edu.uy/plataformas/?p=1.

[4] *Programa del plan Ceibal en Inglés*. URL: https://www.ceibal.edu.uy/es/ceibal-en-ingles.

[5] Cecilia Conti. *La enseñanza de segundas lenguas en la escuela pública y su vínculo con el desarrollo*. URL: http://jornadas.cienciassociales.edu.uy/wp-content/uploads/2016/06/Eje_EducaciÃşn_CeciliaConti.pdf.

[6] *Comunicado de la directora general de Educación Inicial y Primaria en el portal de Presidencia, octubre 2018*. URL: https://www.presidencia.gub.uy/comunicacion/comunicacionnoticias/buzzetti-primaria-escuelas-rurales-ingles.

[7] *Plan Inglés sin Límites, portal de ANEP*. URL: http://www.anep.edu.uy/15-d/primaria-pondr-en-marcha-un-programa-piloto-para-enseanza-ingl-s-en-escuelas-rurales.

[8] *Marco Común Europeo de Referencia para las lenguas equivalencias con los niveles de la Universidad de Cambridge*. URL: https://www.cambridgeenglish.org/es/exams-and-tests/cefr/.

[9] *Acuerdo para certificaciones por parte de Cambridge. Políticas Lingüísticas de ANEP*. URL: http://www.anep.edu.uy/plinguisticas/index.php/cambridge.

[10] *Resultados de las evaluaciones adaptativas de Ceibal en Inglés para estudiantes escolares. Ediciones 2016 y 2017*. URL: https://ingles.ceibal.edu.uy/biblioteca/categoria/primer-categoria-de-biblioteca.

[11] *Material del curso 2018 de Introducción al Procesamiento de Lenguaje Natural, Facultad de Ingeniería, Udelar.*

[12] Anita Ferreira y Gabriela Kotz. «ELE-Intelligent Tutor: A computational parser for the processing of grammatical errors in Spanish as a Foreign Language. Revista Signos.» En: (2010). URL: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-09342010000200002.

[13] Michael Gilleland. «Levenshtein Distance, in Three Flavors». En: (2006). URL: https://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Spring2006/assignments/editdistance/Levenshtein%20Distance.htm.

[14] *NLTK*. URL: https://www.nltk.org/.

[15] *Stanford POS tagger*. URL: https://nlp.stanford.edu/software/tagger.shtml.

[16] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross y Katherine Miller. «Introduction to WordNet: An On-Line Lexical Database». En: (ago. de 1993). URL: http://wordnetcode.princeton.edu/5papers.pdf.

[17] Princeton. *WordNet*. 2019. URL: https://wordnet.princeton.edu/.

[18] Yoav Goldberg y Omar Levy. «Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method». En: 14 (feb. de 2014). URL: https://arxiv.org/pdf/1402.3722.pdf.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado y Jeffrey Dean. «Distributed Representations of Words and Phrases and their Compositionality. En NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2». En: (dic. de 2013). URL: https://arxiv.org/pdf/1310.4546.pdf.

[20] Richard Socher, Christopher Manning y Jeffrey Pennington. «GloVe: Global Vectors for Word Representation. En Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).» En: (oct. de 2014). URL: http://aclweb.org/anthology/D14-1162.

[21] Piotr Bojanowski, Armand Joulin, Edouard Grave y Tomas Mikolov. «Enriching Word Vectors with Subword Information». En: (jul. de 2016). URL: https://arxiv.org/pdf/1607.04606.pdf.

[22] Diane Litman. «Natural Language Processing for Enhancing Teaching and Learning. En Proceeding AAAI'16 Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence». En: (feb. de 2016). URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12310/12208.

[23] Arnaldo Candido, Erick Maziero, Lucia Specia, Caroline Gasperin, Thiago Pardo y Sandra Aluisio. «Supporting the Adaptation of Texts for Poor Literacy Readers: a Text Simplification Editor for Brazilian Portuguese.In Proc. 4th Workshop Innovative Use of NLP for Building Educational Applications.» En: (jun. de 2009). URL: http://aclweb.org/anthology/W09-2105.

[24] Jill Burstein, Jane Shore, John Sabatini, Brad Moulder, Steven Holtzman y Ted Pedersen. «The Language MuseSM System: Linguistically Focused Instructional Authoring». En: (oct. de 2012). URL: https://files.eric.ed.gov/fulltext/EJ1109943.pdf.

[25] Jill Burstein, John Sabatini, Jane Shore, Brad Moulder y Jennifer Lentini. «A User Study: Technology to Increase Teachers' Linguistic Awareness to Improve Instructional Language Support for English Language Learners». En: (jun. de 2013). URL: http://aclweb.org/anthology//W/W13/W13-1501.pdf.

[26] Adriane Boyd, Aleksandar Dimitrov, Vanessa Metcalf y Niels Ott Meurers, W. Detmar Ramon Ziai, Luiz Amaral. «Enhancing Authentic Web Pages for Language Learners. En Proceedings of the NAACL HLT 2010 Fifth International Workshop on Innovative Use of NLP for Building Educational Applications». En: (jun. de 2010). URL: http://aclweb.org/anthology/W10-1002.

[27] Detmar Meurers. «Natural Language Processing and Language Learning». En: (nov. de 2012). URL: http://tjure.sfs.uni-tuebingen.de/~dm/papers/meurers-11.pdf.

[28] Leonardo Zilio, Rodrigo Wilkens y Cédrick Fairon. «Using NLP for Enhancing Second Language Acquisition. En Volume: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017». En: (sep. de 2017). URL: http://acl-bg.org/proceedings/2017/RANLP%202017/pdf/RANLP107.pdf.

[29] Jonathan Brown y Maxine Eskenazi. «Retrieval of authentic documents for reader-specific lexical practice.» En: (2004). URL: http://www.cs.cmu.edu/~jonbrown/publications/InSTIL-2004.pdf.

[30] Tomas Mikolov y Le An Ha. «Computer-aided generation of multiple-choice tests». En: (mayo de 2003). URL: http://www.aclweb.org/anthology/W03-0203.

[31] Jonathan Brown, Gwen Frishkoff y Maxine Eskenazi. «Automatic Question Generation for Vocabulary Assessment». En: (2005). URL: http://aclweb.org/anthology/H05-1103.

[32] Michael Heilman y Noah A. Smith. «Good Question! Statistical Ranking for Question Generation. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics». En: (jun. de 2010). URL: http://aclweb.org/anthology/N10-1086.

[33] Helen Yannakoudakis y Ted Briscoe. «Modeling coherence in ESOL learner texts. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP». En: (jun. de 2012). URL: http://aclweb.org/anthology/W12-2004.

[34] Zahra Rahimi, Diane Litman, Elaine Wang y Richard Correnti. «Incorporating Coherence of Topics as a Criterion in Automatic Response-to-Text Assessment of the Organization of Writing. In Proceedings 10th Workshop on Innovative Use of NLP for Building Educational Applications». En: (jun. de 2015). URL: http://aclweb.org/anthology/W15-0603.

[35] Nitin Madnani, Jill Burstein, John Sabatini y Tenaha O'Reilly. «Automated Scoring of a Summary Writing Task Designed to Measure Reading Comprehension». En: (jun. de 2013). URL: http://aclweb.org/anthology/W13-1722.

[36] Plan Ceibal. URL: https://www.ceibal.edu.uy/es.

[37] Wordsmyth. URL: https://www.wordsmyth.net/.

[38] Wordsmyth Kids. URL: https://kids.wordsmyth.net/we/.

[39] Cambridge. *Diccionario*. URL: https://dictionary.cambridge.org/.

[40] *Wiktionary*. URL: https://www.wiktionary.org/.

[41] Michael Hart. *Project Gutenberg*. 1971. URL: https://www.gutenberg.org.

[42] *Pattern*. URL: https://www.clips.uantwerpen.be/pages/pattern.

[43] *Simple Wikipedia*. 2001. URL: https://simple.wikipedia.org.

[44] Gensim. *Wikicorpus*. URL: https://radimrehurek.com/gensim/corpora/wikicorpus.html.

[45] *Wikipedia dumps*. URL: https://dumps.wikimedia.org/enwiki/latest/.

[46] Kenneth Heafield. *KenLM*. URL: https://github.com/kpu/kenlm.

[47] *Gensim*. URL: https://radimrehurek.com/gensim/.

[48] Zi Yin. «Understand Functionality and Dimensionality of VectorEmbeddings: the Distributional Hypothesis, the Pairwise Inner Product Loss and Its Bias-Variance Trade-off». En: (2018). URL: https://arxiv.org/pdf/1803.00502.pdf.

[49] *Niveles de inglés, British Council*. URL: https://www.britishcouncil.es/ingles/niveles.

[50] *Adaptación del Marco de Referencia Europeo para las lenguas 2016. Ceibal en Inglés*. URL: https://ingles.ceibal.edu.uy/storage/app/uploads/public/5a7/1e9/db7/5a71e9db73773322875637.pdf.