UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Facultad de Ingeniería

Proyecto de Grado

Construcción de herramientas para enseñanza de inglés: generación de preguntas y respuestas

Autores:
Martín Jaime Morón
Joaquín Scocozza Díaz

Supervisores:
Aiala Rosá
Luis Chiruzzo

12 de mayo de 2020

Agradecimientos

En primer lugar, queremos agradecer a nuestras familias y amigos, por el incansable apoyo brindado durante este último año y principalmente a lo largo de toda la carrera.
Un agradecimiento especial para nuestros tutores Aiala y Luis, quienes nos brindaron toda su dedicación y paciencia durante este proceso.
Al grupo de Políticas Lingüísticas de ANEP y en particular a Aldo Rodríguez, con quien tuvimos diferentes reuniones a lo largo del año y siempre se mostró dispuesto a colaborar.
A la directora Martha Gómez y maestros de la Escuela Número 27 de Canelones, quienes muy amablemente nos recibieron y nos permitieron trabajar con la aplicación junto a los alumnos.

Resumen

Actualmente, la enseñanza de inglés en el Uruguay se desarrolla en gran parte del país, gracias a diferentes proyectos y herramientas que así lo fomentan. Sin embargo, llevar a cabo esta tarea no es sencillo dado que hay zonas en las cuales no es posible contar con profesores especializados en la lengua, como es el caso de muchas escuelas rurales. Esto tiene como consecuencia que en estas escuelas los alumnos tengan más dificultades para poder llegar a los niveles ideales de inglés al terminar la educación primaria.

En los últimos años, el grupo PLN de la Facultad de Ingeniería ha trabajado en conjunto con el Programa de Políticas Lingüísticas de ANEP con el objetivo de proveer herramientas que sirvan de apoyo para la universalización de la enseñanza de inglés a nivel de primaria.
Este proyecto busca continuar con el desarrollo de herramientas que se enfoquen en la generación de ejercicios que faciliten el aprendizaje de la lengua para alcanzar el objetivo mencionado.

Utilizando las diferentes técnicas y recursos del Procesamiento del Lenguaje Natural, se logró la implementación de una aplicación en línea con el fin de ofrecer un nuevo recurso a los ya generados en proyectos anteriores, en los cuales se generan juegos y ejercicios de inglés automáticamente con la asistencia del maestro. La aplicación realizada consta de dos secciones principales: una para los maestros, en la cual pueden ingresar un texto en inglés y la herramienta genera automáticamente preguntas y respuestas a partir de dicho texto, y la otra, enfocada en los alumnos con el objetivo de que puedan visualizar y resolver los ejercicios de preguntas y respuestas generados previamente por los maestros.

Durante la realización del proyecto, fue posible realizar diferentes actividades de vinculación, con el objetivo de compartir el trabajo realizado y recibir retroalimentación para continuar mejorando. Entre las actividades realizadas, se destaca la participación en el Foro de Lenguas de ANEP y la visita a una escuela rural de Canelones.

Finalmente, consideramos que el trabajo realizado cumplió con los objetivos planteados al inicio del proyecto. Se logró implementar una aplicación alineada con el objetivo de contribuir con la enseñanza de inglés en las escuelas, facilitando el trabajo de los maestros automatizando la generación y corrección de ejercicios de preguntas y respuestas.

Palabras clave: enseñanza, inglés, PLN, ANEP, generación de preguntas, generación de respuestas, SRL, aplicación.

Índice

Índice de figuras

Índice de cuadros

1. Introducción

Siendo el inglés una de las lenguas más habladas a lo largo de todo el mundo, se considera de suma importancia la enseñanza de este idioma desde los niveles iniciales de la educación.

En la actualidad, esta necesidad en la educación pública del Uruguay, es mitigada mediante el plan "Inglés sin Límites" de ANEP [1], donde los alumnos a partir de cuarto año cuentan con clases de inglés obligatorias. Para ello, las escuelas urbanas cuentan con clases de inglés presenciales y/o remotas por parte del Programa Ceibal en Inglés [2]. Para el caso de las escuelas rurales, la solución mencionada resulta parcial, ya que por motivos de accesibilidad y/o conectividad, no todas las escuelas de este tipo logran acceder a docentes presenciales o videoconferencias con docentes del exterior.

A pesar de que el proyecto ha logrado acercar la enseñanza de esta lengua a las escuelas de todo el país, siguen existiendo centros de estudio (mayormente rurales) donde por carencia de los recursos previamente mencionados, esta no puede ser ofrecida.

El presente trabajo surge como continuación de dos proyectos de grado [3][4] propuestos por el grupo PLN de la Facultad de Ingeniería de UDELAR, realizados a lo largo del año 2018. Ambos tuvieron como finalidad contribuir con nuevos recursos y/o herramientas para la enseñanza de inglés, intentando brindar una alternativa para aquellas escuelas donde la conectividad no es lo suficientemente buena para la realización de videoconferencias. Como resultado final, estos proyectos colaboraron en la generación de ciertos tipos ejercicios de inglés de manera automática, donde la corrección de los mismos se lleva a cabo sin la necesidad del docente.

En la búsqueda de continuar alineados al objetivo de los proyectos mencionados e intentando seguir contribuyendo a la enseñanza de inglés en las escuelas, este trabajo consiste en la realización de un tipo diferente de ejercicio a los previamente implementados: ejercicio de preguntas y respuestas con corrección automática. Como se detalla en el capítulo 3, la solución implementada consiste en una única aplicación web, con secciones específicas para el alumno y el docente. Por un lado, la sección docente permitirá generar ejercicios de preguntas y respuestas de manera automática, partiendo de un texto en inglés y retornando un listado con preguntas

y respuestas editable, que el docente seleccionará para formular un ejercicio. Por otra parte, en la sección de los alumnos, se contará con el listado de ejercicios disponibles, donde el alumno seleccionará uno y podrá visualizar el texto asociado al ejercicio y responder las preguntas que fueron generadas. A medida que va respondiendo las preguntas, irá recibiendo retroalimentación respecto a qué tan acertada es su respuesta a cada momento.

1.1. Objetivo

Dado el aumento en la necesidad de aprender el idioma inglés, surge la necesidad de lograr su enseñanza para los alumnos de todas las escuelas, independientemente de la zona en que se encuentren.

Para satisfacer esta necesidad, es importante hacer llegar a las escuelas recursos que potencien la enseñanza de este idioma, principalmente aquellas como las rurales, que enfrentan otro tipo de dificultades.

Este proyecto tiene el objetivo principal de ayudar a estas escuelas, dejando a su disposición recursos informáticos que hagan posible la enseñanza del idioma, permitiendo a su vez que esta lengua pueda ser enseñada por maestros que no posean un nivel avanzado del idioma, facilitando recursos cuya generación y corrección es automática. Se considera que esto es importante, ya que permitiría a maestros con niveles menos avanzados de Inglés poder participar en la enseñanza de esta lengua en las escuelas.

Por lo mencionado anteriormente, este trabajo busca proveer mediante el uso de técnicas de Procesamiento del Lenguaje Natural y junto al apoyo del Programa de Políticas Lingüísticas de ANEP en colaboración con la Facultad de Ingeniería de la UDELAR, un conjunto de recursos y herramientas tanto para los alumnos como para los docentes, que les dé soporte en la enseñanza de la lengua con especial foco en las escuelas rurales.

En particular, se busca elaborar una plataforma en línea que permita la generación de ejercicios de preguntas para la comprensión de textos, así como ofrecer la corrección de los mismos de forma automática, minimizando la participación del docente.

1.2. Objetivos específicos

De forma alineada a los objetivos de ANEP, teniendo en cuenta las restricciones temporales en cuanto a la duración de proyecto, así como los intereses tanto de la Facultad de Ingeniería como del grupo que llevó a cabo este proyecto, se identificaron los siguientes objetivos específicos:

- Ser protagonistas en la universalización de la enseñanza de segundas lenguas en Uruguay brindando un conjunto de herramientas para servir de apoyo a los maestros que deben enseñar Inglés más allá del lugar geográfico en el que se encuentren y de las limitantes que estos tengan en el Inglés. En particular, se busca que mejore el proceso de aprendizaje de los alumnos de las escuelas públicas del Uruguay, de manera que estos puedan alcanzar los niveles A1 y A2 correspondientes al estándar de niveles de inglés definidos por la Universidad de Cambridge.

- Llevar a cabo la confección de una herramienta que sea accesible para todas las escuelas y que facilite la generación, validación y corrección de ejercicios bajo la supervisión del docente, siguiendo los lineamientos planteados en los trabajos anteriores de la Facultad de Ingeniería en el área descriptos en la sección 2.3.6.

- Crear una aplicación que cumpla con estándares de calidad en cuanto a usabilidad, es decir que sea rápida, intuitiva y siga buenas prácticas con respecto a su diseño. De esta forma, se puede lograr una mejor interacción tanto con los maestros como con los alumnos.

- Hacer efectivo el uso de la aplicación por usuarios reales, tanto de alumnos como maestros, para la validación de la misma así como también para plantear posibles mejoras o trabajos futuros en el área.

- Lograr un conocimiento general respecto a los diferentes avances del Procesamiento del Lenguaje Natural en el área de la educación, así como también del estado del arte en lo que respecta a la generación automática de preguntas y respuestas a partir de textos en inglés.

- Utilizar y comparar diferentes herramientas y técnicas del Procesamiento del Lenguaje Natural para la generación de preguntas y respuestas, así como también para la enseñanza del inglés en general.

1.3. Motivación del Grupo

Fueron varios los motivos que nos impulsaron a trabajar en este proyecto.

Una de las razones principales fue el hecho de trabajar en un proyecto directamente relacionado a la educación, particularmente en niveles iniciales de la misma, donde consideramos es fundamental contar con las herramientas y recursos necesarios para que los estudiantes se encuentren motivados a aprender. Esto también implica que la investigación y el desarrollo realizado podrá ser utilizado para contribuir en un factor social, ya que será utilizado por maestros y alumnos, siendo este un atractivo muy importante para nosotros en el proyecto.

Por otra parte también nos sedujo que el área principal del presente trabajo sea el Procesamiento de Lenguaje Natural, ya que en años anteriores ambos cursamos asignaturas relacionadas a esta temática, donde generamos una fuerte curiosidad e interés en el tema. Además, este campo de las ciencias de la computación ha presentado un crecimiento importante en los últimos años (de la mano con los avances en técnicas de Aprendizaje Automático), por lo cual resulta también atractivo el hecho de profundizar en un área de trascendencia en la actualidad.

1.4. Organización del documento

A continuación se detalla la estructura general de los siguientes capítulos del informe.

En el siguiente capítulo, Marco Teórico, se realiza una breve introducción al Procesamiento de Lenguaje Natural. Se presentan algunos de los problemas más importantes que el área intenta resolver, así como también las herramientas y técnicas utilizadas a lo largo de todo el proyecto. Finalmente, en el capítulo se resumen algunos trabajos relacionados, los cuales sirvieron como base para la realización del proyecto.

En el tercer capítulo, Requisitos y Análisis de la Solución, se hace un planteo de los requisitos generales del proyecto, tanto funcionales como no funcionales. Luego

se desarrolla un esquema general de la solución, que incluye los aspectos básicos de los algoritmos y procesos utilizados en la solución final del proyecto.

En el cuarto capítulo, Desarrollo de la solución, se presentan en detalle todos los aspectos de la solución realizada, junto con el proceso evolutivo realizado para llegar a ella. A su vez, en esta sección se incluyen aspectos específicos de la implementación de cada una de las partes de la solución, junto a las herramientas utilizadas en cada una de estas. Finalmente, el capítulo presenta los resultados de la evaluación realizada y los recursos que fueron generados durante el proceso de implementación.

En el quinto capítulo, Aplicación y Arquitectura, se muestra la aplicación implementada junto con las secciones específicas para cada uno de los tipos de usuario. Luego se procede a explicar en detalle la arquitectura utilizada, así como los diferentes servicios expuestos para lograr la correcta distribución de la aplicación y garantizar su disponibilidad.

En el sexto capítulo, se presentan las actividades de vinculación realizadas durante la realización del proyecto. En particular se realizaron dos reuniones con la contraparte de ANEP, una presentación en el Foro de Lenguas y la visita a una escuela rural con el fin de presentar y validar la aplicación.

En el séptimo capítulo, Conclusiones y Trabajo a Futuro, se detallan las conclusiones obtenidas y aspectos en los que se podría continuar trabajando para mejorar la aplicación.

Por último, los capítulos ocho y nueve presentan el Glosario y la Bibliografía respectivamente.

2. Marco Teórico

Este capítulo tiene como objetivo introducir los conceptos y recursos del Procesamiento de Lenguaje Natural (PLN) utilizados como base para la realización de este proyecto.

En la última sección de este capítulo, se presentará una breve introducción del estado del arte actual para proyectos relacionados con PLN y la generación automática de preguntas y respuestas.

2.1. Procesamiento de Lenguaje Natural

Una de las ramas más importantes de las ciencias de la computación, la lingüística y la inteligencia artificial es el Procesamiento del Lenguaje Natural [5]. La misma tiene como objetivo utilizar la capacidad computacional para procesar diferentes formas de expresión del lenguaje humano (tanto oral como escrito) y lograr la comprensión y generación de lenguaje natural.

La investigación en el área se ha visto impulsada por la gran cantidad de empresas tanto de índole público como privado, con interés en el procesamiento de la información en forma de lenguaje natural disponible gracias a internet. Si bien estas empresas son capaces de retener a la mayoría de los expertos en el área, es la necesidad de mejorar y colaborar lo que llevó a que se desarrollen muchas herramientas de PLN con licencias *open source* de buen nivel y continuar contribuyendo en el área de manera colaborativa.

Entre las diferentes aplicaciones del Procesamiento del Lenguaje Natural se destacan [1]:

- *Traducción Automática*: Generación de algoritmos capaces de traducir de un lenguaje a otro sin la intervención humana. Con esta aplicación se abre la posibilidad de generar contenido accesible para personas que no hablen el idioma original en la que se encuentra cierta información.

- *Reconocimiento del habla*: Generación de texto a partir del habla. Esta aplicación ofrece grandes beneficios, donde se destaca la posibilidad de generar

[1] Se mencionan simplemente algunas, también se podría agregar el resumen automático, inteligencia de marketing, *chatbots*, clasificación de textos, corrección ortográfica, entre otros.

información en formato de texto que al momento solo se encontraba disponible en formato de audio. Es de gran utilidad para aquellas personas que no poseen capacidad auditiva pero sí visual.

- *Análisis de sentimientos*: Clasificación tanto de texto como de habla para la inferencia de sentimientos. Esta aplicación ofrece diferentes usos, entre ellos la posibilidad de analizar grandes volúmenes de datos para obtener información acerca de los sentimientos de las personas sobre determinado tema.

- *Respuesta de Preguntas*: Generación de respuestas a ciertas preguntas en formato de lenguaje natural, que puedan responderse con la información relevada de forma automática. Esta última aplicación es una de las más complejas en el proceso de recuperación de información, dado que se parte de una pregunta en lenguaje natural (forma más común entre los seres humanos al momento de querer informarse acerca de algo).

- *Extracción de Información*: Consiste en la transformación de información des-estructurada embebida en un texto en información estructurada. Por ejemplo, para ser ingresada en una base de datos relacional o para permitir más procesamiento sobre la misma.

A continuación, se detallan los diferentes pasos comunes en el acto del procesamiento del lenguaje natural disponibles en la figura 1.

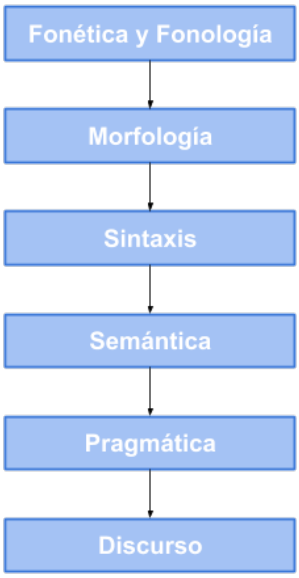

Figura 1: Pasos comunes en el procesamiento del lenguaje natural [6]

Como primer paso se encuentra la fonética y fonología, es decir, el estudio de los sonidos lingüísticos. Luego se realiza un estudio morfológico, de la estructura interna de las palabras. Después, se estudia la sintaxis, orden y agrupamiento de las estructuras de orden mayor, para luego hacer un estudio del significado de las palabras (semántica). Finalmente se realiza un estudio de la pragmática, es decir, de cómo el lenguaje se utiliza para cumplir los objetivos y por último se realiza un estudio de las unidades mayores a la oración (discurso) [7].

2.1.1. Pos Tagging

Part-of-speech tagging es el proceso de asignación de una *tag* (o etiqueta) con información de la categoría gramatical a cada una de las palabras del texto. Como entrada se utiliza una secuencia de palabras y un conjunto de *tags*, y la salida es una secuencia de *tags*, una para cada *token* [5].

En el presente trabajo, se utilizaron las *tags* definidas por el Penn Treebank [8], presentadas en la Figura 2 junto con su correspondiente definición.

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb gerund	*eating*
JJ	adjective	*yellow*	VBN	verb past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, sing.	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Figura 2: Penn Treebank *POS tags*

Un ejemplo de aplicación de POS tagging podría ser el siguiente:
Sea la oración *"John likes the blue house at the end of the street"*, si se aplica el etiquetado utilizando POS tags, podría obtenerse un resultado como el que muestra la figura 3.

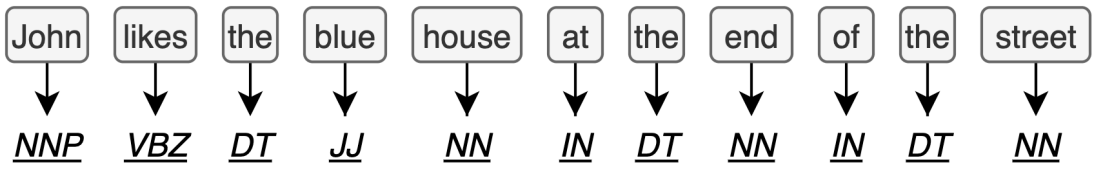

Figura 3: Ejemplo de POS tagging

En el proceso de encontrar el *tag* correcto, se genera un proceso de desambiguación, en el cual se debe elegir entre varios posibles *POS tags*. Un ejemplo de esta desambiguación para el idioma Español es el caso de la palabra *nada*. La misma puede aparecer utilizada tanto como verbo (*nada en la piscina*) así también como pronombre (*nada es gratis*). Este es uno de los problema más complejos que enfrenta el *POS tagging*, ya que si bien solo el 14 % de las palabras en Inglés son ambiguas, estas son muy frecuentemente utilizadas [5].

2.1.2. Dependency Parsing

Existen diferentes enfoques para el análisis sintáctico de las oraciones. En este proyecto se trabaja puntualmente con en Análisis de Dependencias, que se define como el análisis de la estructura sintáctica de una oración, estableciendo relaciones entre palabras *head* y las palabras que modifican estas palabras *head* (*dependant*) [9]. En los últimos años el interés en representaciones de dependencias en el procesamiento del lenguaje natural ha sido motivado por la necesidad de desambiguar las relaciones entre las palabras de un texto [10].

En *Dependency Parsing*, la relación de dependencia asimétrica entre una palabra *head* y una *dependant* se hace relacionando explícitamente ambas, siendo la *dependant* la palabra que depende directamente de la *head*. En la figura 4 se puede apreciar un esquema ilustrativo de un grafo de dependencias de *Dependency Parsing* para la frase "*Michael is ten years old.*".

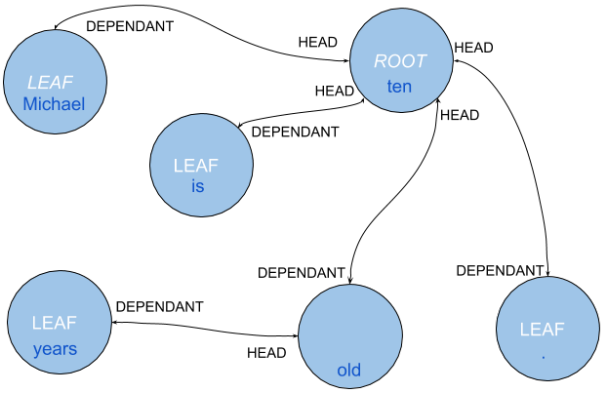

Figura 4: Esquema ilustrativo de un grafo de dependencias de *Dependency Parsing* para la frase "*Michael is ten years old.*"

.

Dependency Parsing también permite la clasificación gramatical de cada una de las relaciones, es decir, le asigna una etiqueta a las relaciones.

Ejemplos de estas etiquetas son: de argumento causal (sujeto nominal, objeto directo, objeto indirecto, etc), modificador nominal (modificador posicional, modificador numérico, etc) entre otros.

Un ejemplo de aplicación de *Dependency Parsing* para la oración *United canceled the morning flights to Houston* se puede apreciar en la figura 5.

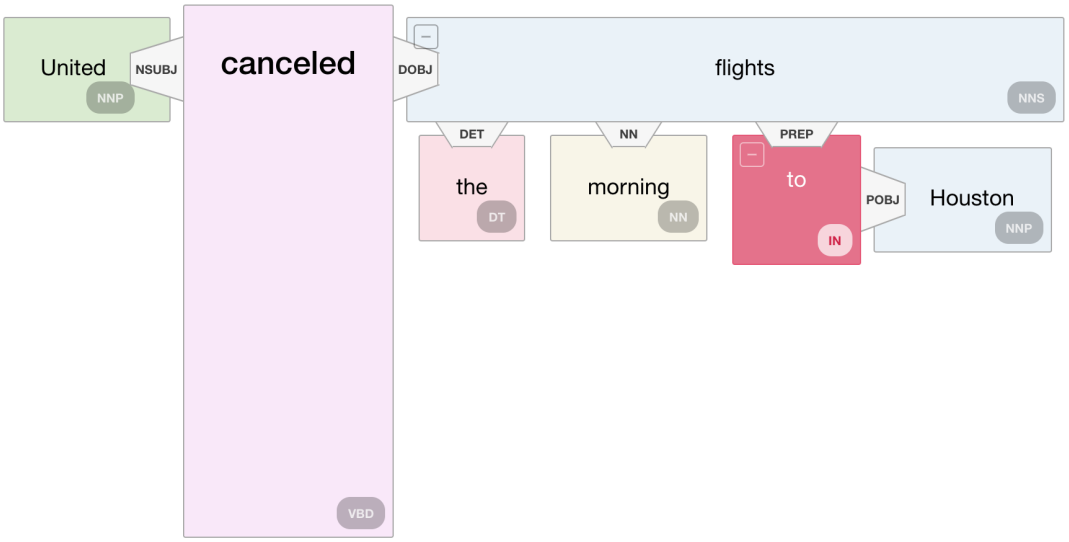

Figura 5: Ejemplo de aplicación de *Dependency Parsing* para la oración *"United canceled the morning flights to Houston"*.

Como se aprecia en la figura, *canceled* es la palabra padre (raíz o *root*), y a partir de ella se desprenden diferentes relaciones de dependencia, sujeto nominal (*Houston*) y objeto directo (*flights*). A su vez *flights* se relaciona con el resto de las palabras como *head*.
El *Dependency parsing* se puede modelar como un grafo en el cual las palabras pueden no estar situadas en los nodos sumideros (sin arcos salientes) del mismo.

En cuanto a los formalismos del *Dependency parsing*, se destacan las siguientes reglas:

- Existe un nodo designado que no tiene arcos hacia él, denominado nodo raíz.
- A excepción del nodo raíz, el resto de los nodos tienen uno y sólo un arco hacia ellos.
- Existe un camino del nodo raíz a cualquiera de los otros nodos, y este camino es único.

2.1.3. Semantic Role Labeling

El rol semántico es un término de la lingüística, también conocido como rol temático, que expresa la función semántica que desempeña un sintagma en la acción o el estado que describe el verbo en la oración [11]. La lista de roles semánticos más estandarizados se puede apreciar en el cuadro 1.

Thematic	**Role Definition**
AGENT	The volitional causer of an event
EXPERIENCER	The experiencer of an event
FORCE	The non-volitional causer of the event
THEME	The participant most directly affected by an event
RESULT	The end product of an event
CONTENT	The proposition or content of a propositional event
INSTRUMENT	An instrument used in an event
BENEFICIARY	The beneficiary of an event
SOURCE	The origin of the object of a transfer event
GOAL	The destination of an object of a transfer event

Cuadro 1: Algunos roles semánticos utilizados comúnmente junto a sus correspondientes definiciones [12]

Se define Etiquetado de Roles Semánticos, o SRL por sus siglas en inglés (*Semantic Role Labeling*) a la tarea de, a partir de cada predicado en una oración, determinar los roles semánticos en la misma.

En la actualidad, la mayoría de los enfoques se basan en aprendizaje automático supervisado. FrameNet y PropBank son los proyectos más utilizados para especificar qué se considera como un predicado, definir el conjunto de roles utilizados en cada tarea y proveer un conjunto de entrenamiento y otro de test [5].

En la figura 6 se presenta un ejemplo de etiquetado de roles semánticos para la oración *"John bought blue shoes"*.

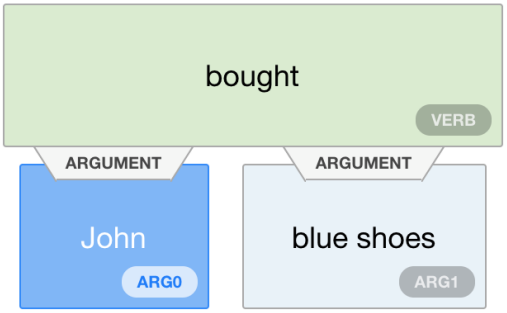

Figura 6: *Semantic Role Labeling* para la oración *"John bought blue shoes"*.

En cuanto a las etiquetas utilizadas para el etiquetado de roles semánticos, Propbank utiliza las siguientes etiquetas:

- *ARG0*: proto-agente, engloba el agente, la fuerza y más.
- *ARG1*: proto-paciente, engloba el tema, el paciente y más.
- *ARG2*: Usualmente denota beneficiario, instrumento, atributo, o estado final.
- *ARG3*: Usualmente denota el punto de partida, beneficiario, instrumento o atributo [2].
- *ARG4*: Usualmente punto de llegada o destino.

Además de estas etiquetas de argumentos, existen otro tipo de etiquetas denominadas de función o modificadoras, entre las que se encuentran ARGM-LOC (modificador de ubicación), ARGM-TMP (modificador temporal), ARGM-DIR (modificador de dirección) y más [13].

En este proyecto, se utiliza la herramienta de SRL que provee AllenNLP (sección 2.2.2). A diferencia de varias herramientas que se basan en algunos estudios previos que sugieren el hecho de que la información sintáctica cumple un papel importante a la hora de etiquetar, AllenNLP desafía esa percepción con modelos neuronales de SRL que logran resultados bastante buenos.

[2]Propbank solamente propone roles estándar para ARG0 y ARG1, a partir de ARG2 su función depende del verbo. Por ejemplo, el hecho de que el beneficiario sea ARG2 o ARG3 depende del verbo.

2.1.4. Named entity Recognition

Al proceso de etiquetar las diferentes entidades que se mencionan en un texto, se le denomina *Named Entity Recognition*. En cuanto a las entidades nombradas, las mismas se definen como cualquier referencia en el texto de algo con nombre propio, como puede ser una organización, un lugar, o una persona [5].

Si bien es un hecho que el término *named entity* no se encuentra definido en su totalidad (existen diferentes interpretaciones de lo que abarca), por lo general se extiende el concepto de *named entity* mencionado antes (organización, un lugar, o una persona) para incluir fechas, horas, otras formas de expresiones temporales, e incluso expresiones numéricas como precios [14].

El proceso en cuestión incluye la clasificación de la entidad nombrada en algunos de los grupos antes mencionados, es decir, organización, lugar, persona, etc.

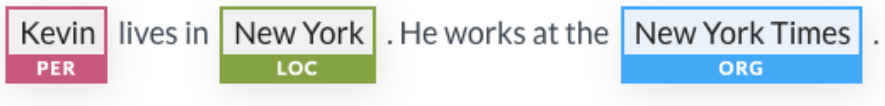

Figura 7: Ejemplo de *Named Entity Recognition* para la oración *"Kevin lives in New York. He works at the New York Times"*

Para el ejemplo de la figura 7, se distinguen tres entidades nombradas:

- *Kevin*: Persona
- *New York*: Ubicación
- *New York Times*: Organización[3]

Este proceso de etiquetado, como muchos en la rama del procesamiento del lenguaje natural, tiene que lidiar con un proceso de desambiguación. A modo de ejemplo, *Louis Vuitton*[4] puede ser catalogado como persona, organización, o producto comercial. En la oración *"Louis Vuitton is in New York"*, si bien se puede determinar que la entidad *Louis Vuitton* en este caso no refiere al producto

[3]New York Times es un reconocido diario de la ciudad de New York.

[4]Empresa francesa de marroquinería de lujo especializada en artículos de viaje fundada por Louis Vuitton.

comercial, se genera la ambigüedad de si refiere a la persona que esté en *New York*, o que sea la organización que esté presente en *New York*.

2.1.5. Coreference Resolution

El objetivo principal de la resolución de correferencias es, a partir de un texto con diferentes entidades a lo largo del mismo, lograr una representación que defina en qué expresiones se está haciendo referencia a una entidad y a cuál. La resolución de correferencias se utiliza en gran parte de los trabajos de alto orden de Procesamiento de Lenguaje Natural, como por ejemplo resumen de textos, respuesta de preguntas y extracción de información.

A modo de ejemplo, con la oración *"John is short. His sister is Sally. She is tall."*, se puede extraer una representación como la de la figura 8 en la cual hay dos entidades: *John* y *Sally*. A su vez, se puede concluir que semánticamente la palabra *His* hace referencia a *John*, mientras que tanto *His sister* como *She* hacen referencia a *Sally*.

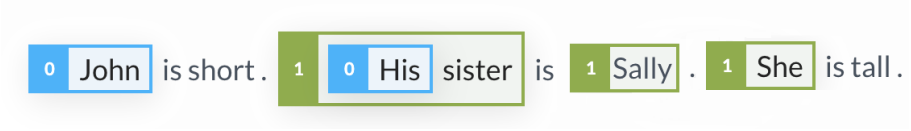

Figura 8: Resolución de correferencias para la oración *"John is short. His sister is Sally. She is tall."*

Las conclusiones que se pueden extraer a partir de la ejecución de este análisis son muy interesantes. En el caso anterior, en base a algún análisis más profundo, se puede interpretar que *John* tiene como hermana a *Sally*. Este dato podría ser utilizado para:

- Responder a la pregunta de quién es la hermana de *John*.
- Resumir el enunciado a que ambos son hermanos.
- Procesar el enunciado para almacenar la relación de hermandad entre ambos.

En el área de la traducción automática, la resolución de correferencias también tiene un papel muy importante [5]. Por ejemplo, al querer traducir del Español al

Inglés la oración « *"Me encanta el conocimiento", dijo*», debe determinarse si debe traducirse como « *"I love knowledge", he said*», o « *"I love knowledge", she said*»[5].

2.2. Técnicas y recursos para el procesamiento de textos

A continuación se realiza una breve introducción teórica a las herramientas del Procesamiento de Lenguaje Natural más utilizadas a lo largo del proyecto.

2.2.1. Distancia de mínima edición

La distancia de mínima edición surge del problema de encontrar la similitud entre dos palabras y definir la noción de similitud lexicográfica entre dos palabras cualesquiera.

Esta técnica es de gran utilidad para la corrección ortográfica. Por ejemplo, si un usuario escribe la palabra "tmate" y se desea determinar lo que intentó comunicar, entre las palabras candidatas está la palabra "tomate", que difiere en una letra y parece intuitivamente mejor candidata que otras palabras que también difieren en una letra.

Ejemplos de palabras similares:

- Hola - Ola
- Roja - Rota
- Lago - Algo

El método que mejor se adapta a la resolución de este problema es el de la programación dinámica, es decir, mediante la utilización de subproblemas superpuestos y subestructuras óptimas [16].

El método original consiste en determinar con cuántas operaciones de edición se puede partir de una palabra y llegar otra.

[5]Este ejemplo proviene de un artículo acerca de una profesora que fue traducida incorrectamente como *"he"* por el traductor de Google Translate debido a la imprecisa resolución de correferencias [15].

F	R	A	N	C	I	A
I	T	A	L	-	I	A
s	s	-	s	i	-	-

Cuadro 2: Distancia de mínima edición entre Francia e Italia

Como se puede apreciar en el cuadro 2, partiendo de la palabra Francia, para llegar a Italia, hay que realizar cuatro operaciones, tres de sustitución y una de inserción.

La descripción formal del problema [7] es:

- Sea X $palabra_1$ de largo n
- Sea Y $palabra_2$ de largo m

Se define una matriz $d(n+1, m+1)$

Se inicializan la fila y columna 0:

- d(0,0) = 0
- Para i de 1 .. n d(i,0) = i
- Para j de 1 .. m d(0,j) = j

Se define la distancia de mínima edición como:

```
Para cada i de 1 a n:
  Para cada j de 1 a m:
      d(i,j) = min(
         d(i, j-1) +1,
         d(i-1, j) +1,
         d(i-1,j-1) + 0 if X(i) = Y(j) si no 2
      )

Retornar d(n,m)
```

La distancia de Levenshtein [17], es una forma de calcular la distancia mínima entre dos palabras siguiendo el algoritmo presentado, pero con costo de inserción 1, borrado 1, y finalmente de sustitución 2.

2.2.2. AllenNLP

AllenNLP es una librería de código abierto desarrollada para el procesamiento de textos basada en métodos de *Deep Learning* [18] y mantenida por ingenieros e investigadores del *Allen Institute for Artificial Intelligence*.

Dado que fue desarrollada con fines de investigación, es una herramienta que posee cualidades que la hacen una opción a tener en consideración a la hora de comenzar un proyecto que requiera herramientas de PLN.

Entre sus cualidades, se destaca su portabilidad dado que utiliza la herramienta Docker (descrita en el capítulo 5.2), es gratuita, tiene repositorios en Github[6] de donde puede ser descargada, posee una documentación clara y por último, está al nivel del estado del arte en muchas de sus herramientas, actualizándose de manera constante y consiguiendo que se publiquen varios artículos interesantes acerca de ella para el área del Procesamiento de Lenguaje Natural.

Por otra parte, la herramienta provee una API (*Application Programming Interface*) muy sencilla, con múltiples opciones de configuración ajustables a medida para cada problema a resolver.

Una de las razones por las cuales se optó por utilizar esta herramienta, es el hecho de que en contraste con muchos otros enfoques, la herramienta fue diseñada para dar soporte a modelos que predicen representaciones de estructuras semánticas del texto introducido, como pueden ser los *clusters* de correferencias [19].

Otro de los motivos que llevaron a utilizar $AllenNLP$, fue el de experimentar con una herramienta diferente a las utilizadas por los proyectos del 2018, donde se utilizó la librería $StanfordNLP$ para tareas similares.

Para la realización de este proyecto, se hace uso de cinco funcionalidades provistas por la librería:

- *Pos Tagging*, sección 2.1.1
- *Semantic Role Labeling*, sección 2.1.3
- *Dependency Parsing*, sección 2.1.2
- *Named Entity Recognition*, sección 2.1.4

[6]GitHub es una empresa global que provee un servicio de almacenamiento y control de versionado del código de proyectos de Software

- *Coreference Resolution*, sección 2.1.5

2.2.3. WordNet

Dentro del Procesamiento de Lenguaje Natural, uno de los mayores desafíos es el estudio de las relaciones del significado de las palabras con el objetivo de lograr un mejor análisis semántico de las oraciones (y de las palabras dentro de estas) con el cual se pueda intentar resolver la ambigüedad de las palabras, es decir, qué significado tiene una palabra dentro de una oración [20].

Muchas veces utilizamos determinadas relaciones léxicas sin ser conscientes de ello. Una de estas relaciones es la de la hiponimia, que denota cuando un lexema es subclase de otro. Por ejemplo "niña" es subclase o hipónimo de "persona". A la relación inversa se la denomina hiperonimia (en el caso anterior "persona" es un hiperónimo de "niña"). Otros ejemplos de relaciones son la meronimia y holonimia, donde la primera denota que una clase tiene parte de otra. La relación inversa es la holonimia. La figura 9 presenta un fragmento de jerarquía de hiperónimos en Wordnet.

Wordnet es una ontología de la Universidad de Princeton [20], este tesauro es una base de datos que representa los sentidos de las palabras, contiene las relaciones léxicas entre estas palabras (ver figura 9), así como también cuenta con una breve definición de cada una de estas.

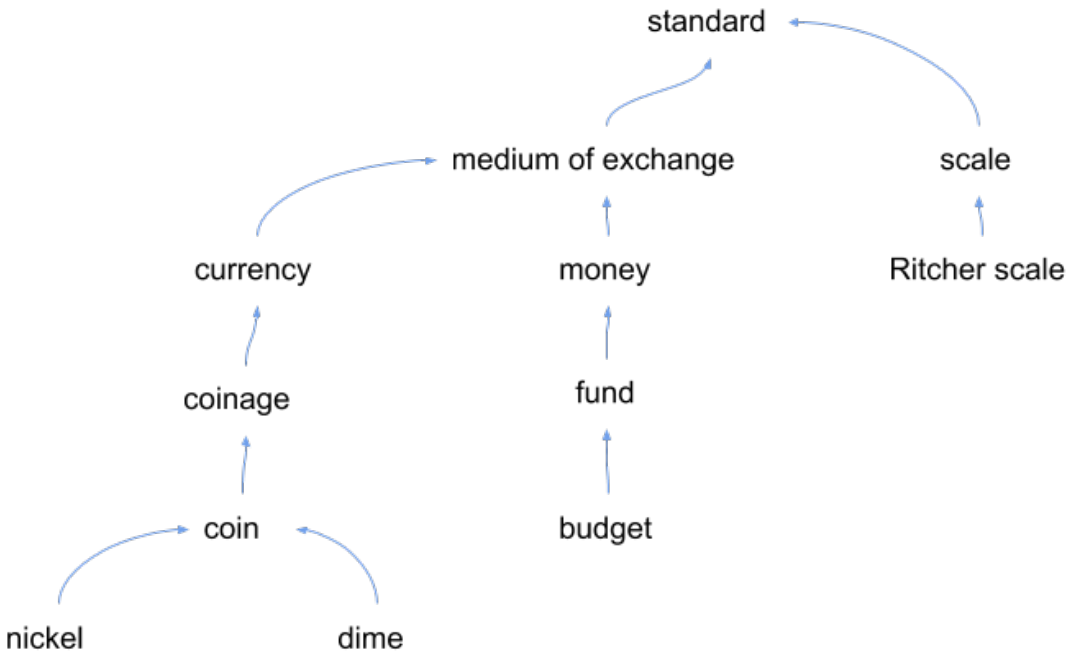

Figura 9: Fragmento de jerarquía de hiperónimos en Wordnet

La unidad mínima de expresión en Wordnet es el *synset* o *set* de sinónimos, palabras sustituibles en al menos un contexto. A modo de ejemplo, para el sustantivo *bag* la base de datos contiene los siguientes *synsets* [22] [21]:

- S: (n) bag (a flexible container with a single opening) *he stuffed his laundry into a large bag.*
- S: (n) bag (the quantity of game taken in a particular period (usually by one person)) *his bag included two deer.*
- S: (n) base, bag (a place that the runner must touch before scoring) *he scrambled to get back to the bag.*
- S: (n) bag, handbag, pocketbook, purse (a container used for carrying money and small personal items or accessories (especially by women)) *she reached into her bag and found a comb.*
- S: (n) bag, bagful (the quantity that a bag will hold) *he ate a large bag of popcorn.*

- S: (n) bag, traveling bag, travelling bag, grip, suitcase (a portable rectangular container for carrying clothes) *he carried his small bag onto the plane with him.*
- S: (n) bag, old bag (an ugly or ill-tempered woman) *he was romancing the old bag for her money.*
- S: (n) udder, bag (mammary gland of bovids (cows and sheep and goats)).
- S: (n) cup of tea, bag, dish (an activity that you like or at which you are superior) *chemistry is not my cup of tea*; *his bag now is learning to play golf.*

Cada uno de los *items* contiene un conjunto de palabras sustituibles en al menos un contexto por definición. Por otra parte, cabe destacar que estas palabras presentan ambigüedad, dado que la misma palabra puede ser usada con diferentes significados.

Como se presentará en la sección 4.1.1, este recurso fue utilizado con el fin de determinar si los adjetivos presentes en una oración corresponden a un color o no, analizando la relación léxica entre estos y la palabra *color*.

2.3. Trabajos Relacionados

En esta sección se describen algunos trabajos en donde se aplican técnicas de PLN para la enseñanza y donde se presentan diferentes enfoques para la generación automática de preguntas y respuestas.

En particular, estos trabajos fueron de gran utilidad para evaluar los diferentes enfoques para la generación y contemplar las ventajas y restricciones de cada uno de ellos.

2.3.1. Natural Language Processing for Enhancing Teaching and Learning

Este trabajo realizado por Diane Litman [23], es un buen punto de partida para introducir la importancia del PLN en la educación y conocer cuáles son los desafíos y problemas al momento de implementar dichas técnicas en la práctica.

El Procesamiento de Lenguaje Natural (PLN), posee más de 50 años de historia como disciplina científica, con aplicaciones a la educación desde 1960. Existen

diversas formas en las que el PLN puede mejorar la tecnología de la educación, de las cuales se mencionan:

- Enseñanza y aprendizaje de materias relacionadas con el lenguaje
- Utilizar el lenguaje para enseñar cualquier asignatura
- Procesar el lenguaje para soportar las necesidades de los estudiantes, maestros e investigadores.

A su vez, también se hace mención a los desafíos que se presentan al momento de utilizar técnicas y recursos del PLN en la educación. Entre ellos, se destacan:

- Desempeño en tiempo real
- Obtener enfoques escalables

2.3.2. A Rule based Question Generation Framework to deal with Simple and Complex Sentences

En la búsqueda de un algoritmo apropiado para la generación de preguntas a partir de textos en Inglés, este trabajo [24] de la Universidad de Jadavpur intenta resolver el problema utilizando un enfoque basado en reglas.

El objetivo planteado en este material es el de resolver el problema de generación de preguntas tanto para oraciones simples como complejas (a diferencia del presente trabajo que se enfoca en textos considerados simples). Para la realización del mismo, se hace uso de los siguientes recursos: VerbNet Corpus, NLTK POS Tagger, NLTK Parser, Stanford NER y Porter Stemmer.

Este trabajo aborda el problema planteado basado en la distinción de dos tipos de preguntas:

- *Factoid* (factoides, sobre hechos): Se generan en base a oraciones simples, y tienen como respuesta un hecho simple.
- *Definition*: Tienen una respuesta "descriptiva", que puede estar distribuida a lo largo del documento.

En términos generales, el algoritmo utilizado en este trabajo consiste en encontrar en cada segmento, oraciones que contengan un grupo nominal seguido de un grupo verbal (referidas como *clause*) y en base a esto inferir qué camino seguir. Descripción de los pasos:

1. Segmentos se delimitan usando comas (,).
2. Cada segmento se tokeniza. Luego con POS Tagger y Parser se ubican las *POS tags* de cada palabra.
3. Se busca en cada segmento por una *clause*. El trabajo brinda una regla específica para esta tarea (identificar *clause*).
4. *Clause* no identificada → Se genera una pregunta reemplazando el segmento y dejando el resto como está.
5. Si hay *Clause* → Vuelve al paso (3).

Para evaluar las reglas implementadas en este trabajo, se utilizaron 2 *datasets*. El *dataset*1 contiene 220 oraciones simples, para el cual se prepararon manualmente un total de 454 preguntas posibles. Por otro lado, el *dataset*2 contiene 70 oraciones simples y complejas con un total de 264 preguntas diferentes. Los cuadros 3 y 4 presentan un resumen de los resultados obtenidos para el trabajo en cuestión.

Tipo de Pregunta	**Incorrectas**	**Ambiguas**	**Aceptables**
Who	1	29	145
What	15	42	153
Whose	0	0	5
How Much	1	4	17
How Many	1	2	9
Whom	0	0	38
Where	2	0	8
When	1	0	2

Cuadro 3: Resumen de los resultados obtenidos para el *dataset*1.

Tipo de Pregunta	Incorrectas	Ambiguas	Aceptables
Who	4	20	71
What	24	16	98
Whose	1	0	9
How Much	1	0	5
How Many	2	0	11
Whom	1	0	5
Where	18	0	18
When	0	0	11

Cuadro 4: Resumen de los resultados obtenidos para el *dataset*2.

2.3.3. Good Question! Statistical Ranking for Question Generation

Uno de los problemas enfrentados por nuestra solución fue el de la excesiva generación de preguntas para textos relativamente cortos (por ejemplo se podrían llega a generar alrededor de 20-30 preguntas, en un texto de simplemente 2 párrafos).

El trabajo en cuestión [25] intenta brindar un enfoque estadístico para la mitigación de este problema, ofreciendo las preguntas generadas utilizando un *Ranking*. Para lograr esto, se utiliza un proceso en dos etapas para la generación de preguntas:

1. *Simplificación del Texto*: Alterando elementos léxicos, estructuras sintácticas y semánticas. El conjunto de transformaciones aplicadas, derivan en una forma más simple del texto original.

2. *Transformación en Preguntas*: A la salida obtenida en 1, se le aplican transformaciones léxicas y sintácticas, para formar un conjunto de preguntas (se logran preguntas con: *what*, *when*, *who*, *where* y *how much*).

Una vez obtenido el conjunto de preguntas generadas y con el fin de poder ordenarlas, se procede a calificar las preguntas de acuerdo a qué tan aceptables son. El criterio de aceptabilidad para cada pregunta se refleja en las $features$ utilizadas por la función de *Ranking*. Entre estas, se menciona el largo de la pregunta (cantidad de *tokens*) y las transformaciones sintácticas aplicadas a la oración para su generación, entre otras.

Debido a que no se dispone de grandes cantidades de datos del dominio, estilo y forma adecuada de las preguntas, la calificación de las preguntas se aprende a partir de un *dataset* relativamente pequeño de preguntas generadas por el algoritmo y etiquetado por humanos.

En cuanto a resultados obtenidos, la métrica utilizada para evaluar la función de *Ranking* fue el porcentaje de preguntas generadas marcadas como aceptables para el primer N % del *Ranking*, para varios valores de N. Mientras que el 27.3 % de todo el conjunto de preguntas de prueba fue aceptable, 52.3 % del primer 20 % en el *Ranking* fue aceptable. Por lo tanto, la calidad del primer quinto (20 %) de preguntas en el *Ranking* fue casi duplicado por el *Ranking* utilizando todas las $features$.

2.3.4. Machine Learning Approach to the Process of Question Generation

En este trabajo [26] se plantea el uso de *Machine Learning* para la generación de preguntas y respuestas a partir de textos no estructurados de forma de colaborar con la enseñanza (intentando contribuir con recursos que no requieran la supervisión directa de un docente).

La principal ventaja de este enfoque radica en la posibilidad de mejorar gradualmente el sistema agregando más datos a través del $feedback$ recibido en cada uso (se agregan más pares pregunta-oración con $feedback$ positivo o negativo para las preguntas ya generadas). Sin embargo, esta misma capacidad también se puede ver como una desventaja, ya que para lograr buenos resultados se requiere disponer de grandes volúmenes de datos.

Para la evaluación de este trabajo, se entrenó el sistema con un *dataset* de 306 artículos acerca de países obtenido de Wikipedia. El *dataset* fue dividido en dos partes: artículos con nombres que comienzan con las letras A a la D, aproximadamente el 25 % de los artículos, para entrenamiento y el resto para la evaluación. En los cuadros 5 y 6 se presenta un resumen de los resultados obtenidos por este trabajo, donde la correctitud de las preguntas generadas fue evaluada manualmente.

Dataset	Artículos	Oraciones	Preguntas Generadas	Correctitud (%)
Entrenamiento	51	499	1098	91,8
Evaluación	155	1540	1466	87,9

Cuadro 5: Resumen de los resultados obtenidos para cada *dataset*.

Tipo de Pregunta	Total	Correctas	Incorrectas	Correctitud (%)
What	701	628	73	89,6
True-False	429	398	31	92,8
Where	186	162	24	87,1
How many	39	38	1	97,4
Who	62	46	16	74,2

Cuadro 6: Resumen de los resultados obtenidos por tipo de pregunta para el *dataset* de evaluación.

2.3.5. Generating Questions Automatically from Informational Text

Este trabajo presenta un enfoque basado en reglas (o *templates*[7]) similar al presentado en 2.3.2, donde se utilizan recursos diferentes como base para la generación [27]. Se utilizan diferentes plantillas con las posibles preguntas a generar (manteniendo siempre una estructura base), implementadas a través de reglas.

En particular, el trabajo se enfoca en textos informativos (siendo esta la continuación de un trabajo para textos narrativos), cuya principal motivación viene dada por generar preguntas que validen el entendimiento de cierto texto por parte de los estudiantes. Para cumplir este objetivo, se intenta generar preguntas de forma de que el estudiante se cuestione y reflexione sobre lo que ha leído. Por ejemplo, dada la oración: *"All goats should have covered shelters where they can escape the weather"*. Una posible pregunta sería: *"Why should goats have covered shelters?"*

En total, el trabajo agrega los siguientes 4 *templates* de preguntas, que se alinean con el objetivo de fomentar la reflexión en los estudiantes:

- *What would happen if [x] ?*

[7]Algunos trabajos diferencian el enfoque de reglas y el de templates, argumentando que las reglas que se implementan siguiendo un enfoque de templates son más rígidas en cuanto a las posibles formas que pueden tomar las preguntas generadas. A efectos del presente trabajo, ambos enfoques se mencionarán de manera indistinta.

- *When would [x]?*
- *What happens [temporal-expression]?*
- *Why [auxiliary-verb] [x]?*

A modo de ejemplo, para el caso del *template* para preguntas de tipo *Why*, la etiqueta [x] del *template* se mapea con los roles semánticos ARG0, TARGET, ARG1 y ARG2 (en caso de haber).

Si se observa la oración del ejemplo, la información utilizada para generar la pregunta de tipo *Why* es:

- *[auxiliary-verb] = should*
- *[x] = [ARG0 goats] [VERB have covered] [ARG1 shelters]*

Para los demás *templates*, también se utiliza el etiquetado de roles semántico con el fin de obtener la información requerida por los diferentes *templates*.

Para evaluar la calidad de los resultados obtenidos, se contempló que la pregunta generada sea gramaticalmente correcta y tenga sentido en el contexto del texto. El cuadro 7 presenta un resumen de las estadísticas y resultados obtenidos en este trabajo.

Tipo de Pregunta	**Preguntas Generadas**	**Porcentaje de Preguntas Correctas**
Condition	15	86.7 % (13/15)
Temporal Information	88	65.9 % (58/88)
Modality	77	87.0.9 % (67/77)

Cuadro 7: Resumen de resultados obtenidos para "Generating Questions Automatically from Informational Text"

2.3.6. Proyectos anteriores

En los últimos años, la Facultad de Ingeniería en colaboración con ANEP ha trabajado en varios proyectos enfocados en la generación de ejercicios de inglés de forma supervisada por los docentes. Estos proyectos son:

- *Construcción de herramientas para soporte a la enseñanza de lenguas* [3].

- *Aplicaciones Lúdicas de Soporte a la Enseñanza de Lenguas* [4].

En ambos proyectos, se logró realizar aplicaciones que sirvieran para la generación de ejercicios para la enseñanza de inglés. En particular, estos trabajos se focalizan en poder lograr la universalidad de la enseñanza de segundas lenguas, teniendo como desafío principal que los docentes de algunas escuelas no necesariamente sean expertos en estas.

Ambos trabajos se enfocan en la construcción de estas aplicaciones utilizando herramientas y técnicas de Procesamiento de Lenguaje Natural.

Entre los recursos y técnicas utilizados por ambos trabajos se encuentran: *POS tagging*, Análisis de constituyentes, Análisis de dependencias, Extracción de información, Extracción de definiciones, *Word Sense Desambiguation*, Modelado de Lenguaje, *Word Embedding*, *Word2vec* y más.

Ambas aplicaciones generadas cuentan con un conjunto de juegos y ejercicios que el profesor puede generar para que los alumnos utilicen. Entre ellos, se encuentran los presentados en las figuras 10, 11, 12 y 13.

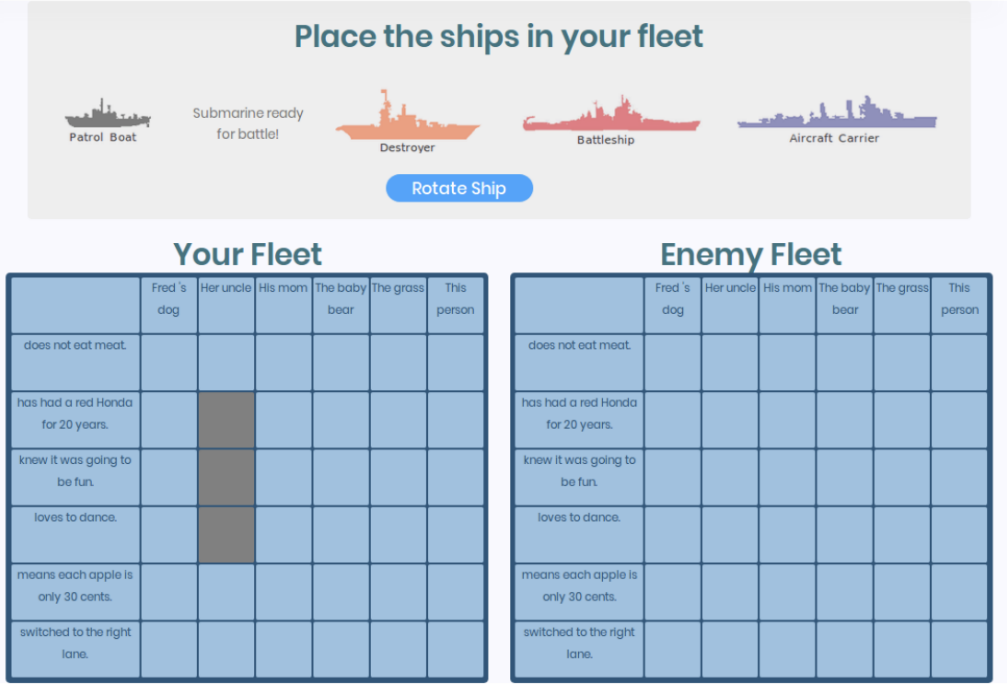

Figura 10: Batalla Naval. Del trabajo: Aplicaciones Lúdicas de Soporte a la Enseñanza de Lenguas[4]

Figura 11: Sopa de letras. Del trabajo: Aplicaciones Lúdicas de Soporte a la Enseñanza de Lenguas[4]

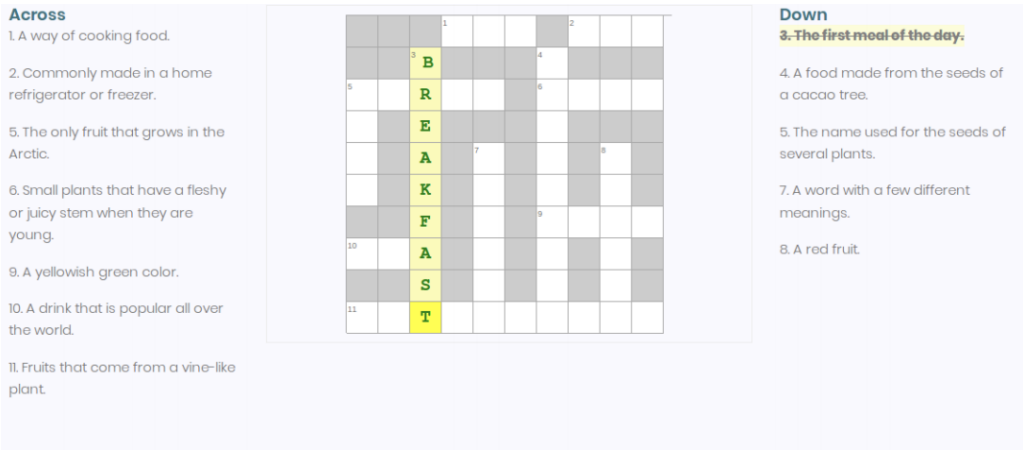

Figura 12: Crucigrama. Del trabajo: Aplicaciones Lúdicas de Soporte a la Enseñanza de Lenguas[4]

Figura 13: Completar oración. Construcción de herramientas para soporte a la enseñanza de lenguas[3]

3. Requisitos y Análisis de la Solución

El presente capítulo tiene como objetivo introducir aspectos generales de la solución que se presentará en profundidad en el siguiente capítulo.

Por un lado se detallarán los requerimientos relevados con los que debe cumplir la solución, y a continuación se introducirá la solución que se propone en alto nivel.

3.1. Requerimientos relevados

Dada la complejidad del proyecto y de las distintas entidades que participan en el mismo, fue necesario realizar un análisis inicial de los requisitos antes de comenzar con la implementación de la solución.

Por este motivo, se mantuvieron diferentes reuniones con cada uno de los involucrados, de forma de lograr clarificar los requisitos para cada uno de los posibles usuarios finales, es decir, para los estudiantes, docentes y administradores.

3.1.1. Requisitos funcionales

El principal requisito a cumplir por parte de nuestra solución, es la posibilidad de que un docente pueda ingresar a la aplicación y generar un ejercicio de preguntas y respuestas automáticamente a partir de un texto ingresado.

Otro requisito que se relaciona de manera directa con el anterior, es que luego de que el conjunto de preguntas y respuestas fueron generadas de manera automática, el docente debe contar con la posibilidad de:

- Agregar una pregunta/respuesta
- Editar una pregunta/respuesta
- Eliminar una pregunta/respuesta
- Asignarle un nombre a la propuesta

En cuanto a los alumnos, el requisito principal de la aplicación es que estos puedan acceder a la aplicación, seleccionar un ejercicio planteado por los docentes, y resolverlo.

A este último requisito se le añade la posibilidad de que una vez que el alumno se encuentra resolviendo un ejercicio, se debe ofrecer retroalimentación en cuanto

a qué tan acertada es su respuesta. Dado que el objetivo de este tipo de ejercicios es el de valorar la comprensión del texto por parte del estudiante, la respuesta no tiene que ser perfecta desde un punto de vista ortográfico y sintáctico para ser evaluada como correcta.

Se acordó considerar como correctas a todas aquellas respuestas que contengan todas las palabras de alguna de las posibles respuestas. Por ejemplo, para una pregunta que tenga como respuesta la palabra *red*, se considerarían correctas las respuestas *It's red*, *Is red* o incluso alguna que contenga errores ortográficos e incluya la palabra *red* (entre otras).

En cuanto a los requisitos de administrador, se consideró de gran valor la posibilidad de visualizar el conjunto de ejercicios generados (conjunto de preguntas con sus respectivas respuestas), así como las respuestas que los alumnos brindaron a dichas preguntas.

3.1.2. Requisitos no funcionales

De forma de poder lograr una aplicación de fácil acceso, con buena usabilidad y alineada con los criterios de calidad esperados, es que se relevaron diferentes requisitos no funcionales.

En primer lugar, debe ser posible acceder a cualquier parte de la aplicación desde una "ceibalita" limitando el uso de internet al mínimo (debido a los posibles problemas de conectividad que pueden existir). Dadas las diferentes capacidades de procesamiento que poseen las computadoras (limitada en el caso de las ceibalitas), para poder hacer uso de las diferentes herramientas de PLN utilizadas de forma eficiente, se decidió que no necesariamente deba ejecutarse la generación del ejercicio en la ceibalita, sino que el mismo pasa a generarse en el servidor.

En cuanto a requisitos en lo que respecta al tiempo de respuesta de la aplicación, se planteó flexibilidad en cuando al tiempo en la generación de un ejercicio (ya que es una tarea vital en nuestra solución), pero el mismo no debe ser excesivo. Para evaluar este requerimiento, se estableció que el tiempo máximo aceptable (en promedio) para la generación para textos típicos para el nivel de inglés esperado, no debe superar los 20 segundos. Como es de suponer, el tiempo de respuesta depende del tamaño del texto ingresado. Por este motivo se realizaron diferentes pruebas con textos de diferente tamaño considerados pertinentes para este tipo de

ejercicios a este nivel.

En lo que respecta a la seguridad, el requisito principal es que los usuarios de administrador sean protegidos por usuario y contraseña. Además, la generación de los ejercicios debe estar al menos protegida por contraseña compartida (entre docentes) de forma que los alumnos no puedan generar ejercicios.

Otro requisito no funcional relevado fue la restricción de que la complejidad de los ejercicios generados debe ser acorde a los niveles de inglés A1 y A2. Cabe destacar que esto también dependerá del nivel de inglés del texto ingresado para generar las preguntas.

Como último requisito relevado, se planteó el hecho de que la aplicación debe ser fácil de utilizar. Tanto los profesores como los alumnos, deben ser capaces de utilizar la aplicación sin necesidad de capacitación adicional. La navegación en la aplicación debe ofrecer un nivel de respuesta alto, es decir, la transición entre pantallas y/o secciones no debe ser mayor a 3 segundos. De esta forma, se logra mantener la atención de los alumnos y ofrecer retroalimentación en tiempo real (un usuario promedio de una aplicación web generalmente abandona el sitio si luego de 10 segundos no obtiene respuesta) [28].

3.2. Análisis de la Solución

Para poder cumplir con los objetivos planteados se procedió a implementar una única aplicación web, con secciones específicas para docentes y alumnos.

En la sección del docente, el usuario procede a introducir un texto en inglés, luego la aplicación genera de forma automática una lista de preguntas con sus respectivas respuestas. A partir de las preguntas y respuestas generadas por el sistema, el docente tiene la posibilidad de editar las respuestas, agregar respuestas, y por último elegir las preguntas que considera más adecuadas. En términos de usabilidad, una primera aproximación de estas funcionalidades puede apreciarse en la figura 14, donde se muestra la sección docente para la generación de un ejercicio.

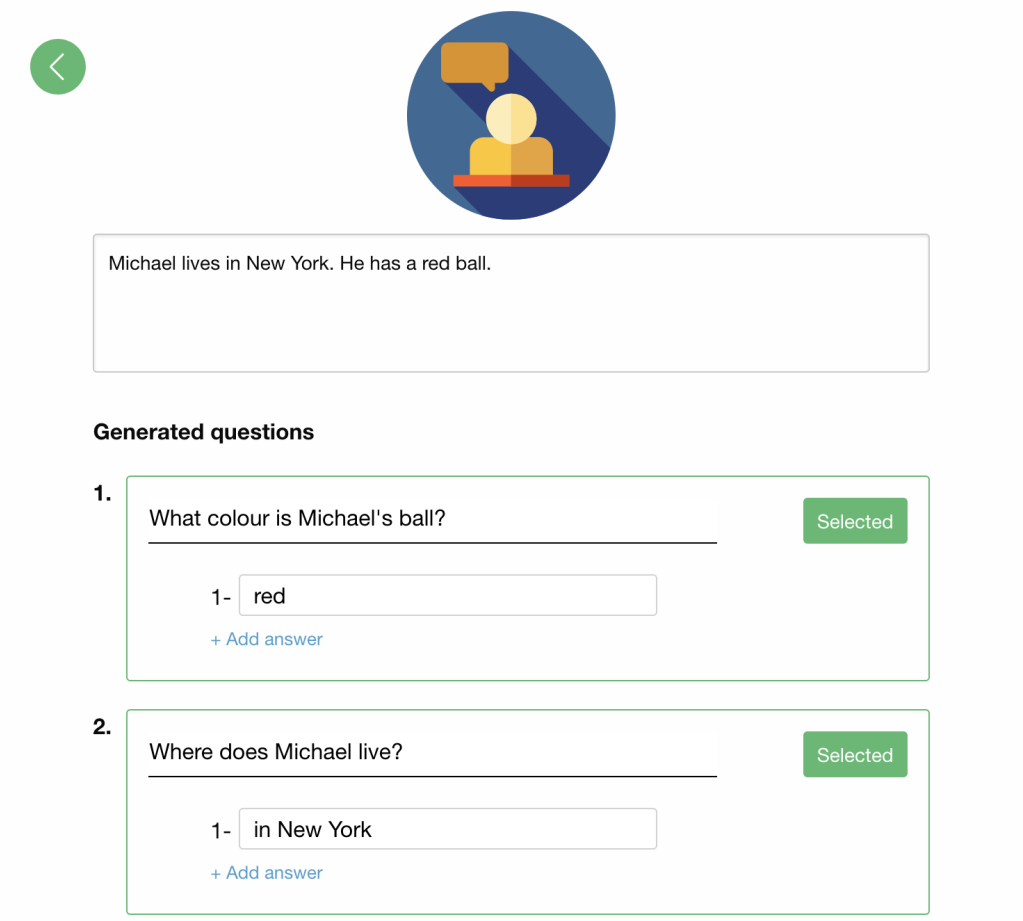

Figura 14: Sección Docente - Generación de Preguntas y Respuestas

Por otra parte, en la sección de estudiantes (la cual puede ser accedida tanto por los docentes como por los alumnos) se encuentra una lista con todos los ejercicios generados hasta el momento con el título correspondiente.

Una vez que el alumno elige el ejercicio que desea resolver, se despliegan las preguntas generadas para el mismo, junto con el texto correspondiente para que el alumno pueda responder. A medida que el alumno responda las preguntas, irá recibiendo retroalimentación constante indicando que tan bien están sus respuestas. La figura 15 presenta la interfaz diseñada para la resolución de un ejercicio en la sección de estudiantes.

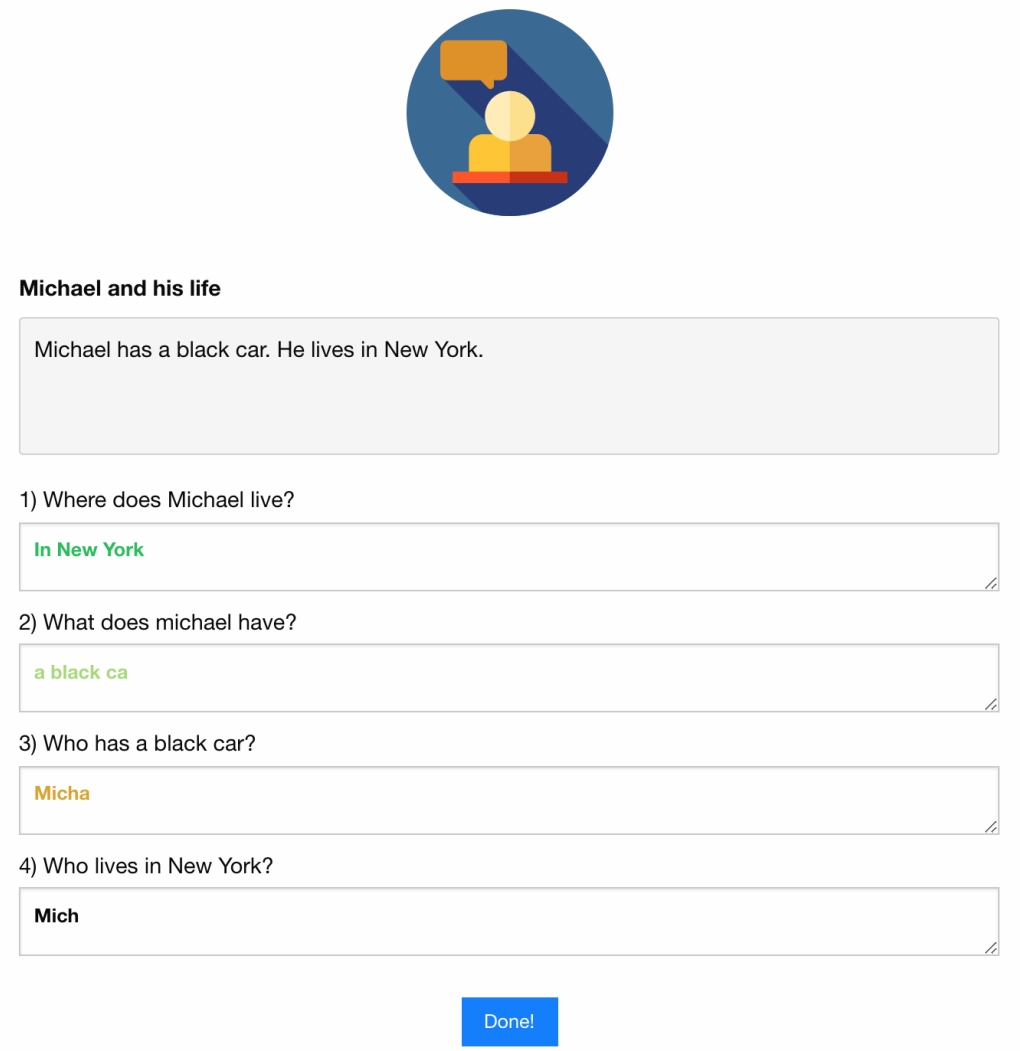

Figura 15: Sección de estudiantes para la resolución de un ejercicio

Finalmente, cuando el estudiante considera como finalizado el ejercicio, se muestra en pantalla el resultado obtenido con un algoritmo que compara las respuestas planteadas por el docente con las introducidas por el alumno.

3.3. Generación de preguntas

Como fue presentado previamente en la sección 2.3, existe más de un enfoque posible para la generación automática de preguntas. Dentro de los utilizados con

mayor frecuencia, se destacan:

- Enfoque basado en Reglas o Templates
- Enfoque basado en Aprendizaje Automático

Cada uno de estos enfoques requiere la utilización de diferentes recursos para su implementación. En base a las herramientas disponibles y las necesidades del problema a resolver, se optó por utilizar un enfoque basado en reglas.

Para el caso del enfoque basado en aprendizaje automático, el recurso indispensable para utilizar este método es un corpus etiquetado de preguntas generadas. Dado que al comenzar el proyecto no se contaba con este recurso (generarlo es extremadamente costoso) y que además los resultados observados en las investigaciones que siguen este enfoque fueron similares al resto, se optó por descartar esta opción.

Por otro lado, también existen enfoques basados en reglas o templates, donde se utilizan reglas para identificar la posibilidad de generar una determinada pregunta (ejemplo en sección 2.3.5). En este caso, el recurso principal que se debe generar es el conjunto de reglas para transformar un determinado texto de entrada en un conjunto de preguntas y respuestas. Se consideró que la implementación de estas reglas era la mejor opción, dado que el nivel de inglés esperado por los usuarios es básico (niveles A1 - A2), lo cual implica que la implementación de estas reglas sea en principio una tarea más sencilla de realizar en comparación con niveles más avanzados de dicho idioma. Además, se cuenta con textos y ejercicios para el nivel de inglés, lo cual facilita la tarea de identificar las reglas que se deben implementar en base a las preguntas más frecuentemente utilizadas y permite tener una primera referencia al momento de evaluar la solución.

En cuanto a los tipos de preguntas generados por la solución, se implementaron reglas para generar preguntas de tipo:

- *Who ...?*
- *Where ...?*
- *When ...?*
- *What ...?*

- *What colour ...?*

Donde cada tipo de pregunta se genera con su correspondiente respuesta, obtenida directamente del texto ingresado. Los detalles de implementación de cada una de estas preguntas y ejemplos de cada uno de estos tipos, serán presentados en la sección 4.1.

3.4. Procedimiento para la Generación de Preguntas

La estructura de la solución se compone de varios módulos relacionados entre sí, de modo que a partir de un cierto texto en inglés, se logre procesarlo y obtener el conjunto ordenado de preguntas y respuestas. El esquema general de la solución propuesta se puede apreciar en la figura 16.

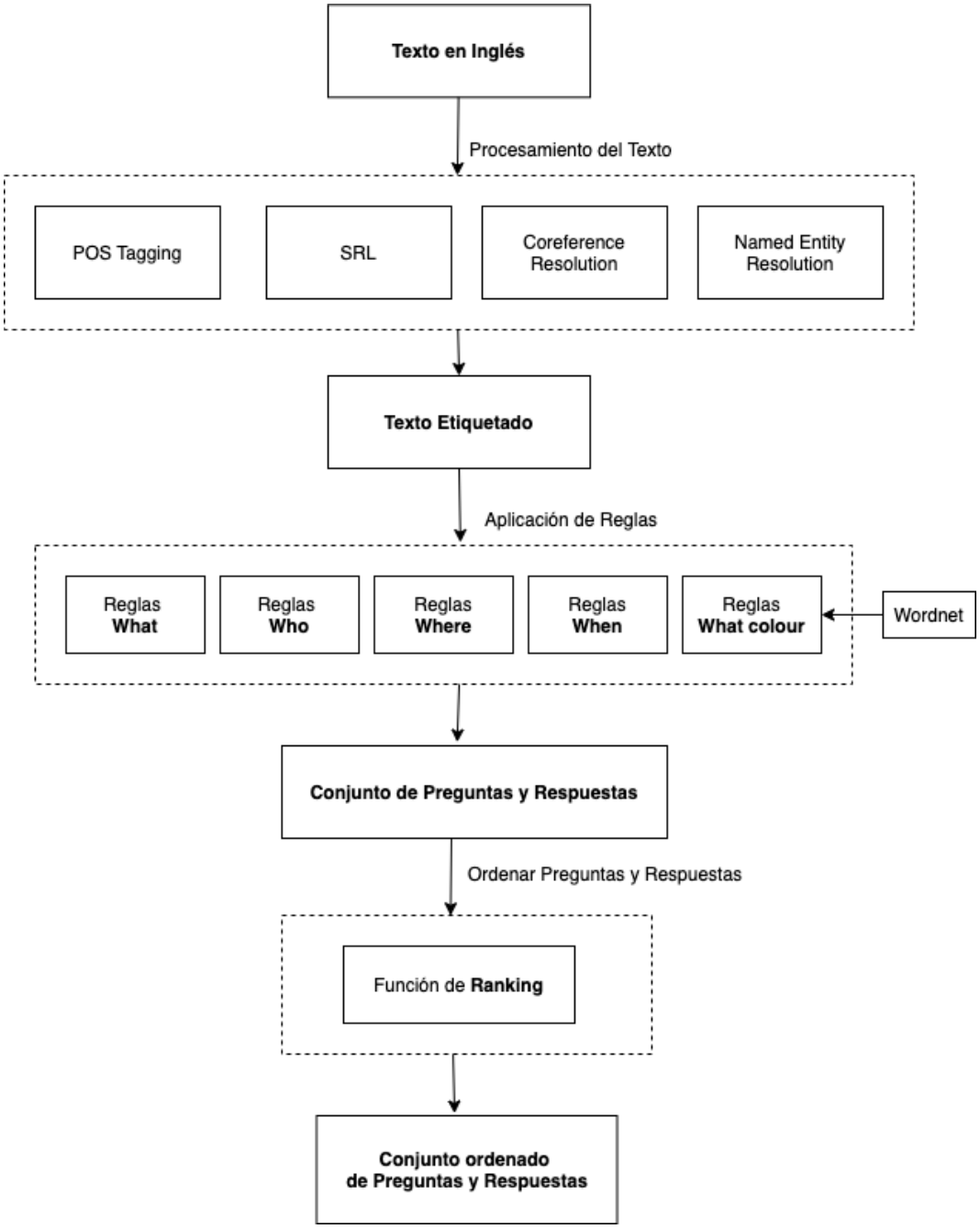

Figura 16: Procedimiento para generación de preguntas

El algoritmo para la generación de preguntas puede explicarse en 3 etapas principales:

1. Procesamiento del texto de entrada, con el fin de obtener la información necesaria para la aplicación de las reglas.

   El resultado de este procesamiento inicial consiste en los siguientes 4 componentes:

   a) Etiquetado Morfosintáctico del Texto (*POS Tagging*)

   b) Etiquetado de Roles Semántico (SRL)

   c) Resolución de Correferencias (*Coreference Resolution*)

   d) Etiquetado de Entidades Nombradas (*Named Entity Recognition*)

2. Procesamiento por parte de los módulos que contienen las reglas, utilizando como entrada la información resultante del paso anterior. En cada módulo se generan las preguntas y sus correspondientes respuestas del texto.

3. Aplicación de la función de *Ranking* (sección 4.2) al conjunto de preguntas y respuestas generado en la etapa anterior. El objetivo principal de esta función es brindar un subconjunto ordenado de preguntas y respuestas al usuario, con el fin de facilitar una sugerencia inicial de formulación de ejercicio.

4. Desarrollo de la Solución

La presente sección tiene como objetivo introducir en detalle la solución implementada, detallando características específicas de la solución y decisiones que debieron llevarse a cabo durante el proceso de desarrollo.

4.1. Reglas

El objetivo de esta sección es presentar en detalle la especificación de los diferentes tipos de reglas utilizados en la generación de preguntas.

Debido a que todas las reglas tienen como alcance la oración, el paso inicial para la generación de todos los tipos de preguntas consiste en la segmentación del texto en oraciones. Además, todas las reglas requieren del etiquetado de roles semánticos (SRL), el etiquetado gramatical (*POS Tagging*) y la resolución de correferencias (*Coreference Resolution*) para su aplicación.

Con el fin de mantener los algoritmos breves y claros, estos pasos serán omitidos en la explicación de cada tipo de regla.

4.1.1. Preguntas What colour

Generación de preguntas de la forma: *What colour ...?*, cuya respuesta siempre contiene al menos un color en la oración.

Ejemplos de preguntas y respuestas de esta forma:

- *What colour is Mary's ball? It is red.*
- *What colour was the sky yesterday? It was grey.*
- *What colour will be the table tomorrow? It will be brown.*

Este tipo de preguntas fue abordado en primer lugar, como prueba de concepto. Si bien puede parecer que son una subclase de las preguntas *what*, que se mostrarán en la sección 4.1.3, las reglas implementadas para la categoría anterior no contemplan este tipo de preguntas, donde surge la necesidad de identificar en la oración un color utilizado como adjetivo (respuesta a la pregunta).

Dado que en los ejercicios de preguntas y respuestas observados en los niveles iniciales de inglés [29] es frecuente encontrar preguntas de esta forma, se consideró importante la definición de reglas específicas que abarquen esta categoría.

En cuanto a la implementación de estas reglas, se presenta a continuación el algoritmo utilizado para las reglas en cuestión:

```
preguntas = []
lista_de_colores = obtener_colores(oracion)

Para cada verbo en la oracion:
  Para cada color en la lista de colores:
    Si el verbo es del tipo to_have:
      Si el color esta en la etiqueta ARG1:
        verbo conjugado = obtener_verbo_conjugado(to be)
        resolver correferencias en ARG0
        pregunta = 'What colour' + verbo conjugado + ARG0 + 's + sustantivo + ?

    Si el verbo es del tipo to_be:
        Si el color esta con la etiqueta ARG2:
            resolver correferencias en ARG1
            pregunta = 'What colour ' + verb + ARG1 + ?

    Si no:
          Si el color esta etiquetado como ARG1 o ARG2:
              verbo_conjugado = obtener_verbo_conjugado(to be)
              pregunta = 'What colour ' + verbo conjugado + 'the' + sustantivo + ?

Retornar preguntas
```

Como se puede apreciar, el algoritmo solamente brinda una noción general de como se generan las reglas de este estilo de preguntas. Por este motivo, a continuación se explicará con mayor detalle algunos aspectos de la implementación, considerados relevantes a la hora de entender el procesamiento realizado.

Obtener Colores

En primer lugar se hace referencia a la función *obtener_colores()*, utilizada para obtener (en caso de existir) el o los colores utilizados en la oración que se está procesando.

Para alcanzar este objetivo, y a su vez con el fin de experimentar con herramientas como Wordnet (2.2.3), se implementó esta función verificando para cada palabra de la oración las siguientes condiciones:

1. La palabra a verificar no está en ARG0 (protoagente). Esto se verifica para evitar formular preguntas cuya respuesta es demasiado obvia, donde la misma puede aparecer formando parte del causante de la acción. Por ejemplo:

Para la oración: *"The white horse is eating"*.
Se generaría: *"What colour is the horse?"*

2. La palabra es un adjetivo, es decir, como resultado del etiquetado gramatical su etiqueta correspondiente es JJ (ver 2.1.1).

3. La palabra es efectivamente un color.

Para determinar si la palabra recibida es un color o no, se utiliza la librería *Wordnet* de *NLTK* (especificada en 2.2.3) [20] [21].

Esta última tarea fue implementada verificando para cada uno de los *synsets* de la palabra, si alguno de estos aparece como hipónimo de *color*. En un principio se verificaba solamente por el primer *synset* de la palabra, pero esto no contemplaba algunos casos como el de la palabra *white*, donde el primer *synset* no es hipónimo de *color*. Dado que esto implica un mayor costo computacional, a modo de mejora se podrían acotar los *synsets* del color para los cuales buscar los hipónimos.

Cabe destacar que este procesamiento para determinar si una palabra es un color o no, podría haberse evitado definiendo una lista de colores (acotados para el nivel de inglés), y verificando simplemente si la palabra pertenece a dicha lista. Con el fin de experimentar y conocer otras herramientas y recursos de PLN, se optó por seguir el enfoque utilizado.

Discriminación según Verbo

Como se puede apreciar, el algoritmo diferencia 3 formas de generar la pregunta según sea el verbo de la oración.
Los casos son:

1. Verbo *"to have"*:
   En este caso, el verbo principal de la oración es el verbo *"to have"*. Por ejemplo, la oración *"John has brown eyes"* genera:
   Question: *"What colour are John's eyes?"*
   Answer: *"brown"*

   La presencia de este verbo, determina que si el color encontrado se encuentra con la etiqueta de proto-paciente en la oración (ARG1), la misma pueda generarse de la siguiente manera:

'What colour ' + verbo *to be* conjugado + ARG0 + 's + sustantivo + ?

En el ejemplo anterior, tenemos:

- verbo *to be* conjugado = are
- ARG0 = John
- sustantivo = eyes

2. Verbo *"to be"*:
   Caso en el cual la oración procesada tiene como verbo principal el verbo *"to be"*. Por ejemplo, la oración *"The ball is red"* genera:
   Question: *"What colour is the ball?"*
   Answer: *"red"*

   Para el verbo "*to be*", la aplicación de roles semánticos determina que el paciente está en ARG2 y el agente en ARG1. Por lo tanto, si se verifica que el color está en ARG2, la pregunta se genera:
   'What colour ' + verbo *to be* + ARG1 + ?

   En el ejemplo anterior, tenemos:

   - verbo *to be* = is
   - ARG1 = The ball

3. Otro verbo:
   Para el resto de los verbos que pueden aparecer como núcleo de la oración, se determina si el color aparece en ARG1 o ARG2 (potencial paciente para el verbo). En caso de que el color provenga de una de estas opciones, se pasa a generar la pregunta de la siguiente manera:
   'What colour ' + verbo *to be* conjugado + 'the ' + sustantivo + ?

   Donde el verbo *to be* es conjugado en el mismo tiempo verbal que el núcleo de la oración y el sustantivo es el adyacente al color obtenido.

Ejemplo: “Mary is ten years old. She washes her red car every Saturday”.
Question: “What colour is the car?”
Answer: “red”

Vale la pena mencionar que dado que las reglas para generar preguntas del tipo “What colour”, fueron implementadas de modo experimental y realizadas al comienzo de la etapa de implementación, las mismas no contemplan todos los tiempos verbales, y están mayormente acotadas al nivel de inglés esperado por los alumnos. Los mejores resultados en la generación de estas preguntas fueron observados en los tiempos verbales: “Present Simple”, “Past Simple”, “Present Continuous” y “Simple Future”.

4.1.2. Preguntas Who

En esta sección se describe cómo se implementó la generación de preguntas que comienzan con *Who*. Este tipo de preguntas son uno de los tipos de preguntas más conocidos y aplicados en las pruebas de los niveles A1 y A2.

Ejemplos de preguntas y respuestas generadas por las reglas en cuestión son:

- *“Who has blue eyes? John”*
- *“Who has a ball? Mary”*
- *“Who played basketball? Tom and Ann”*

La presencia del agente (ARG0) en la oración, refleja que la oración potencialmente puede generar preguntas del tipo *Who*, cuya respuesta será dicho agente.

Por ejemplo, dada la oración:

- Sentence: *“John went to the zoo yesterday”.*
- SRL: [ARG0: John] [V: went] [ARG4: to the zoo] [ARGM-TMP: yesterday]

A continuación se presenta el algoritmo para la generación de preguntas y respuestas de tipo *who*:

```
# NER = Name Entity Resolution
# SRL = Semantic Role Labelling
preguntas = []

Para cada verbo en la oracion:

    Si hay ARG0 o ARG1:
      Obtener el tiempo verbal del verbo
      ARG = ARG0 si ARG0 existe, si no ARG1

      Si es una entidad nombrada (por resolucion de correferencias):
        Si la etiqueta es 'PER' (NER lo califica como persona):
            comienzo_pregunta = 'Who'
        Si no:
          comienzo_pregunta =  'What'
      Si no:
        Si es del tipo 'the' o 'it' o 'that':
          comienzo_pregunta =  'What'
        Si es del tipo 'she', 'he', 'them', 'they':
          comienzo_pregunta = 'Who'

    pregunta = comienzo_pregunta + resto_oracion (sin ARG)
    respuesta = ARG
    preguntas += (pregunta, respuesta)

Retornar preguntas
```

Un ejemplo de aplicación del algoritmo, es para el caso en el que se detecte que la entidad nombrada y que sea un nombre, como en el caso siguiente:

- *"Mike has been sleeping for hours"*
- SRL: [ARG0: Mike] has been [V: sleeping] [ARGM-TMP: for hours]
- NER: [PER: Mike] has been sleeping for hours

Aplicando la regla para la generación de la pregunta, se tiene:

- ARG = Mike (ARG0 en este caso)
- comienzo_pregunta = Who (Mike etiquetado como PER)
- resto_pregunta = has been sleeping for hours
- answer = Mike (ARG0)

Generando así, la siguiente combinación de pregunta y respuesta:
Pregunta: *"Who has been sleeping for hours?"*
Respuesta: *"Mike"*

4.1.3. Preguntas What

Este tipo de preguntas se refiere a todas aquellas que comienzan con la palabra *What*, a excepción de las preguntas de tipo "What colour", que fueron definidas dentro de una sección específica.

Algunos ejemplos de preguntas y respuestas en esta sección son:

- *"What will Mike play tomorrow? football"*
- *"What is under the table? a ball"*
- *"What was red? the wall"*

Estas reglas permiten generar una gran cantidad de preguntas, dado que el indicador principal para la generación de este tipo de preguntas, es la presencia del agente y el paciente asociados al verbo (etiquetas ARG0 y ARG1 respectivamente como resultado del etiquetado de roles semánticos).

Por ejemplo, dada la oración:

- Sentence: *"Mike plays football everyday"*
- SRL: [ARG0: Mike] [V: plays] [ARG1: football] [ARGM-TMP: everyday] .

La presencia de las etiquetas ARG0 y ARG1 es el indicador de que potencialmente se puede generar una pregunta de este tipo.

A continuación se presenta el algoritmo (en términos generales) que implementa las reglas en cuestión:

```
preguntas = []

Para cada verbo en la oracion:

    Si (ARG0 y ARG1 en las etiquetas):
      Obtener tiempo verbal para el verbo
      Conjugar verbo to do en tiempo verbal correcto

      Si (tiempo verbal en [simple present, simple past]):
        verbo_infinitivo = obtener infinitivo (verbo)
        pregunta = 'What ' + to do conjugado + ARG0 + verbo infinitivo + ?

      Si no, Si (tiempo verbal en [
          past continuous, present continuous,
          present perfect, past perfect
      ]):
        pregunta = 'What ' + palabras[posicion_verbo - 1] + ARG0 + verbo + ?

      Si no, Si (tiempo verbal en [
          present perfect continuous, past perfect continuous
      ]):
        pregunta = 'What' + palabras[posicion_verbo - 2] + ARG0
                  + palabras[posicion_verbo - 1] + verbo + ?

      Si no, Si (tiempo verbal en [simple future, future perfect]):
        pregunta = 'What will' + ARG0 + verbo + ?

      Si no, Si (tiempo verbal en [future continuous]):
        pregunta = 'What will'+ ARG0 + be + verbo + ?

      Si no, Si (tiempo verbal en [future perfect continuous]):
        pregunta = 'What will' + ARG0 + 'have been' + verbo + ?

      preguntas += resolver correferencias(pregunta = pregunta, respuesta = ARG1)

Retornar preguntas
```

Como ejemplo de aplicación del algoritmo, se presenta el análisis para la siguiente oración:

- *"Mike will play football"*
- SRL: [ARG0: Mike] [V: will] [ARG1: play football]

Dado que las etiquetas ARG0 y ARG1 se encuentran presentes en el resultado del SRL y el tiempo verbal de la oración es *Simple Future*, corresponde aplicar la siguiente regla:

- *What will + ARG0 + verbo + ?*

En el ejemplo presentado, tenemos que:

- ARG0 = $Mike$
- ARG1 = $football$
- verbo = $play$

Finalmente, aplicando la regla anterior se obtiene la siguiente combinación de pregunta y respuesta:

Pregunta: *"What will Mike play?"*
Respuesta: *"football"*

4.1.4. Preguntas When

En esta sección se presentan las reglas utilizadas para la generación de las preguntas *when*, donde siempre se obtiene como respuesta un indicador de tiempo (ARG-TMP).

Algunos ejemplos de preguntas y respuestas en esta sección son:

- *"When will Tom sign the documents? Tomorrow"*
- *"When did Sally go to the doctor? Yesterday"*
- *"When did Mary live in Madrid? In 1997"*

El indicador principal para la generación de este tipo de preguntas, es la presencia de un modificador temporal, reflejando que la oración incluye información de tiempo para el verbo.

Por ejemplo, dada la oración:

- Sentence: *"Mary goes to school at 8:00 AM"*
- SRL: [ARG0: Mary] [V: goes] [ARG4: to school] [ARGM-TMP: at 8:00 AM]

Donde la presencia de la etiqueta ARGM-TMP, es el indicador de que potencialmente se puede generar una pregunta de este tipo.

A continuación se presenta el algoritmo (en términos generales) que implementa las reglas en cuestión:

```
preguntas = []
Para cada verbo en la oracion:

    # Cuando no es un verbo auxiliar
    Si (ARGM TMP en etiquetas y (ARG0 o ARG1) en etiquetas):
      Obtener tiempo verbal (verbo)
      ARG = si existe ARG0, ARG0, si no ARG1

      argumentos extra = argumentos extra, modificadores, etc

      Si (tiempo verbal en [simple_present, simple_past]):
        Si (es verbo en infinitivo (verbo):
          pregunta = 'When ' + verbo + ARG + argumentos extra + ?
        Si no:
          Obtener verbo to do conjugado en tiempo verbal
          Obtener infinitivo de verbo
          pregunta = 'When ' + to do conjugado + ARG + verbo infinitivo
                     + argumentos extra + ?

      Si no, Si (tiempo verbal es simple_future):
          pregunta = 'When ' + palabras[posicion verbo - 1] + ARG + verbo
                     + argumentos extra + ?

      Si no, Si (tiempo verbal en [present_continuous, past_continuous]):
          pregunta = 'When ' + palabras[posicion verbo - 1] + ARG + verbo
                     + argumentos extra + ?

      Si no, Si (tiempo verbal en [
            future_continuous, future_perfect,
            present_perfect_continuous, past_perfect_continuous
      ]):
          pregunta = 'When ' + palabras[posicion verbo - 2] + ARG
                     + palabras[posicion verbo - 1] + verbo + argumentos extra + ?

      Si no, Si (tiempo verbal en [present_perfect, past_perfect]):
          pregunta = 'When ' + palabras[posicion verbo - 1] + ARG + verbo
                     + argumentos extra + ?

      Si no, si (tiempo verbal es future_perfect_continuous):
          pregunta = 'When ' + palabras[posicion verbo - 3] + ARG
                     + palabras[posicion verbo - 2] + palabras[posicion verbo - 1]
                     +  verbo + argumentos extra + ?

      Si hay pregunta:
        preguntas += resolver correferencias (
                        pregunta = pregunta, respuesta = ARGM TMP
                     )
Retornar preguntas
```

Como se puede apreciar en el algoritmo, se definen reglas para la generación de preguntas en base al tiempo verbal que presente la oración. Es decir, las reglas para las posibles preguntas a generar se determinan en base a la conjugación del verbo. Por ejemplo, para el caso en el que se detecte que el tiempo verbal es "simple future", como en el caso siguiente:
"Bill and Mary will play tennis next week"
SRL: [ARG0: Bill and Mary][ARGM-MOD: will][V: play][ARG1: tennis][ARGM-TMP: next week]
Y dada la regla para la generación de la pregunta, se tiene:

- palabras[posición verbo -1] = will
- ARG = Bill and Mary (ARG0 or ARG1)
- verbo = play
- argumentos extra = "
- respuesta = next week (ARGM-TMP)

Generando así, la siguiente combinación de pregunta y respuesta:
Pregunta: *"When will Bill and Mary play tennis?"*
Respuesta: *"Next week"*

4.1.5. Preguntas Where

En esta sección se presentan las reglas utilizadas para la generación de las preguntas *Where*, donde la respuesta obtenida es siempre un lugar.

Ejemplos de preguntas y respuestas de este estilo son:

- *"Where does Mary play basketball? At school"*
- *"Where was Sally yesterday? In her house"*
- *"Where has Tom and Polly been? At the cinema"*

Similar a como fue presentado para las preguntas *when*, el indicador utilizado en este caso para la generación de este tipo de preguntas, es la presencia de un

modificador de lugar (ARGM-LOC), reflejando que la oración incluye información de lugar para el verbo.

Por ejemplo, dada la oración:

- Sentence: *"Mary plays basketball at school"*
- SRL: [ARG0: Mary] [V: plays] [ARG1: basketball] [ARGM-LOC: at school]

Donde la presencia de la etiqueta ARGM-LOC, es el indicador de que potencialmente se puede generar una pregunta de este tipo.

En términos del algoritmo utilizado, el caso es análogo al de las preguntas *when*, en este caso en lugar de un modificador temporal se utiliza la etiqueta ARGM-LOC que indica la presencia de un modificador de lugar.

4.1.6. Resolución de Correferencias

Como parte de los requisitos relevados en reuniones con los representantes de ANEP (6.1), se solicitó que las preguntas generadas automáticamente no sean solo aquellas cuya respuesta se deriva directamente del texto, sino que también se generen aquellas donde sea necesario procesar e interpretar el texto en su totalidad para poder responder correctamente.

Cuando en una oración aparece una entidad y luego se hace referencia a la misma, aparece la oportunidad de preguntar información que esté asociada a la referencia y esperar que el alumno responda con la entidad. Un ejemplo de esta situación puede ser: *"Michael likes sports. He likes football."*.

En principio, a partir de la primera oración se podría generar la pregunta *Who likes sports?*, donde la respuesta se deriva directamente de la oración que generó la pregunta. Por otra parte, a partir de la segunda oración se podría preguntar *Who likes football?*. En este caso, el estudiante debe resolver a qué entidad hace referencia el pronombre *He* dentro del texto, ya que la respuesta correcta no es *He* sino *Michael*.

En otras palabras, el problema a resolver es: *Dada una lista de palabras, un índice de inicio y un índice de fin, encontrar en la lista entre ambos índices el nombre de la entidad a la cual se hace referencia.*

Para resolver este problema se utilizó el modelo de *Coreference Resolution* de la librería AllenNLP. En términos de implementación, la librería AllenNLP retorna

una lista de pares de índices, donde cada lista representa las referencias a una misma entidad y los índices el lugar de la referencia.

Por ejemplo, para el caso anterior el modelo retorna $0 : [(0, 0), (4, 4)]$ (ver figura 17) lo que se interpreta como una sola entidad, a la cual se hace referencia en los índices del 0 al 0 (Michael) y del 4 al 4 (He).

Figura 17: Resolución de correferencias para: "*Michael likes sports. He likes football.*"

Como un primer acercamiento a la resolución de este problema, se podría pensar que una vez obtenidos los pares de índices, la primera aparición de la referencia será siempre la entidad, lo cual es equivocado.

Un ejemplo concreto en el que esto no se cumple es: "*The owner of the ball is ten, he likes sports. His name is Michael. Michael has two balls.*", en este caso la entidad no es nombrada en la primer referencia. Para poder contemplar este caso y otros similares es que se introdujo el modelo de *Named Entity Recognition* de la librería *AllenNLP*.

La librería retorna en este caso una lista con un conjunto de etiquetas asociadas al *token* en ese índice. En el caso de "*Michael likes sports. He likes football.*", el algoritmo retorna entonces $[$"$U-PER$","O","O","O","O","O","O","O"$]$ (ver figura 18).

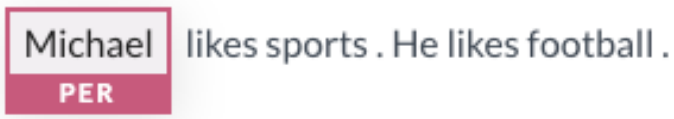

Figura 18: Etiquetado de entidad nombrada para: *"Michael likes sports. He likes football."*

Este esquema responde a UGTO [30].

Las palabras que no refieren a una entidad son etiquetadas como O. Dentro de las palabras que sí refieren a una entidad, se las asigna con U si tiene un POS tag de NNP, T si tiene un POS TAG de LOC MISC u ORG, y en cualquier otro caso se las asigna G. En nuestro caso, basta con saber que la etiqueta no es O para poder inferir que el *token* que se encuentra en ese índice es una entidad nombrada. Volviendo al ejemplo, la entidad nombrada se encuentra en el índice 0.

Para poder encontrar las entidades nombradas dados los índices de las referencias, se implementó el siguiente procedimiento:

1. *Resolución de correferencias*: El primer paso consiste en resolver las correferencias que ocurren el texto.

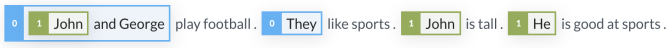

Figura 19: Ejemplo de resolución de correferencias

   La figura 19 presenta un ejemplo de resolución para la oración:
   "John and George play football. They like sports. John is tall. He is good at sports".

   Del ejemplo de la figura 19 se puede concluir que existen dos entidades que aparecen con correferencias[8]. La primera, *John*, se encuentra referenciada en los índices $(0, 0)$, $(8, 8)$, $(11, 11)$. La segunda, *John and George*, se encuentra referenciada en los índices $(0, 2)$, $(5, 5)$.

2. *Reconocimiento de entidades nombradas*: El siguiente paso consiste en realizar el etiquetado de entidades nombradas (NER) en el texto. Al ejecutar este método provisto por la librería *AllenNLP* para el ejemplo anterior, se obtiene el resultado presentado por la figura 20.

[8]La librería *AllenNLP* retorna todas aquellas referencias a entidades si y sólo si estas aparecen referenciadas más de una vez. Por este motivo en el ejemplo de la figura 19 no se indica a *George*, dado que no hay ninguna referencia a dicha entidad (aparte de su primer aparición). Por otra parte, si se marcan *John* (que es referenciado un total de 3 veces), y la pareja *John and George* (que es referenciada un total de 2 veces).

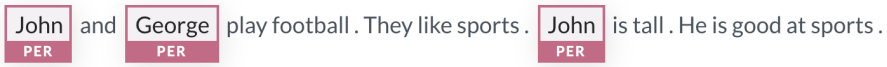

Figura 20: Etiquetado de entidad nombrada

Del ejemplo, se puede concluir que existen tres entidades nombradas. *John* aparece referenciado dos veces, una vez en la posición $(0, 0)$ y otra en la $(8, 8)$, mientras que *George* aparece referenciado una sola en la $(2, 2)$.

3. *Primer mapeo de las referencias a entidades nombradas:* Para cada agrupación de correferencias se busca alguna entidad que esté nombrada exactamente en el mismo lugar que alguna de las referencias, es decir, para el caso de *John and George*, referenciados en las tuplas $(0, 2)$, $(5, 5)$ no existe una entidad nombrada en los mismos lugares. Por otra parte, para el caso de *John*, $(0, 0)$, $(8, 8)$, $(11, 11)$ sí existe, tanto en la $(0, 0)$ como en la $(8, 8)$ por lo que se deduce que cualquiera de las referencias en $(0, 0)$, $(8, 8)$, $(11, 11)$ hacen referencia a la entidad nombrada *John*.

4. *Segundo mapeo de las referencias a entidades nombradas:* Para los casos no resueltos, como el de *John and George*, se procede a buscar dentro de la primer ocurrencia (posición $(0, 2)$) todas las referencias o entidades nombradas dentro de esta y se intenta resolver el problema con los datos ya obtenidos de los casos resueltos.

   En este caso, en la posición $(0, 0)$, se puede saber que hace referencia a *John* ya sea porque existe una referencia en la posición $(0, 0)$ que ya fue resuelta a la entidad *John*, como también se puede resolver por el hecho de que hay una entidad nombrada en exactamente esa posición.

   Luego en la posición $(1, 1)$ no se encuentra ninguna referencia ni entidad nombrada, por lo que se mantiene la palabra en esa posición (*and*). Finalmente, para la posición $(2, 2)$ hay una entidad nombrada exactamente en esa posición, *George*, por lo que se concluye que la entidad *John and George* es referenciada en las posiciones $(0, 2)$, $(5, 5)$.

5. *Resultado final:* Por último, si se desea preguntar para este caso *"Who like sports?"*, y contamos con el índice de la respuesta, en este caso $(5, 5)$, se

obtiene que la respuesta correcta es la entidad nombrada la cual se referencia en esa posición, la cual es *John and George*.

4.2. Ranking de Preguntas

Una de las problemáticas enfrentadas en la generación de preguntas y respuestas, es la cantidad indeterminada de preguntas que potencialmente podrían generarse a partir de cierto texto. A pesar de que una vez que se generan las preguntas, el usuario puede elegir cuáles desea agregar al ejercicio, también resulta en un problema de usabilidad la necesidad de recorrer una lista extensa de preguntas y respuestas para conformar un ejercicio. Con el fin de mitigar esta problemática, se optó por implementar una función de ranking, en la cual como resultado se ofrece al usuario una preselección automática de las 5 preguntas mejor posicionadas en el ranking.

Para la realización de esta función, se definieron 2 criterios básicos para determinar la relevancia (valor en el ranking) de una pregunta para el ejercicio:

1. En primer lugar se tiene en cuenta la posición en el texto de la oración que generó la pregunta. Es decir, se registra qué enunciado del texto generó la pregunta, con el fin de evitar que las preguntas resultantes en el ejercicio provengan de la misma sección del texto.

2. Por otro lado, también se considera la cantidad de preguntas que se generaron para cada tipo de pregunta, teniendo mayor ponderación aquellas para las cuales hay menor cantidad.

A su vez, cabe destacar que la cantidad de preguntas que serán preseleccionadas para el ejercicio (actualmente 5), está definida como una variable y eventualmente podría modificarse o ser determinada por el usuario que utiliza la aplicación.

A continuación se procederá a comentar y analizar los aspectos que se considera son más relevantes de la implementación realizada. En una primera iteración, la función de ranking recibe la totalidad de preguntas generadas por tipo de pregunta (*what*, *what colour*, *where*, *when* o *who*), así como el número de la oración que generó la pregunta. A partir de esta información, se procede a inicializar los valores de la variable *info_qtypes* con la información de las preguntas por tipo. Una vez realizada esta inicialización, se procede a iterar como se muestra a continuación:

```
# Se ordenan las preguntas aleatoriamente para evitar darle
# prioridad a las que provienen de los primeros enunciados.
random.shuffle(questions_list)
while (info_qtypes['total_selected'] < options['max_questions']
    and info_qtypes['total_selected'] < len(questions_list)):
    update_ranking_values(questions_list, info_qtypes, options)
    select_question(questions_list, info_qtypes, options)
```

En cada iteración, se selecciona una nueva pregunta para el ejercicio y se vuelve a calcular los valores del ranking, dado que los valores para el ranking se determinan en base al conjunto de preguntas seleccionado al momento de la iteración. El método *update_ranking_values* actualiza el valor de cada pregunta en el ranking. Para ello, utiliza el siguiente algoritmo para realizar el cálculo del ranking (*rank_value*) para una cierta pregunta dada:

```
rank_value = 1
    # Criterio (1)
    * (1 / (total_position_questions + total_position_selected))

    # Criterio (2)
    * (1 / (total_type_questions + total_selected))
```

Donde:

- total_type_questions = Número de preguntas generadas del tipo actual
- total_selected = Número de preguntas ya seleccionadas del tipo actual
- total_position_questions = Número de preguntas generadas para la posición actual.
- total_position_selected = Número de preguntas seleccionadas para la posición actual.

Como se puede apreciar, los dos criterios básicos mencionados anteriormente se reflejan en el cálculo del valor del ranking por medio de dos factores. En términos de implementación cada factor es opcional, es decir, por medio de variables puede

elegirse qué criterio utilizar para el cálculo del ranking (por defecto ambos se utilizan).

Finalmente, el último paso de la iteración consiste en la ejecución de *select_question*, que simplemente se encarga de agregar a la preselección, la pregunta (y respuesta) con el mayor valor de ranking. Una vez se alcanzaron las 5 preguntas (*max_questions*) o hay menos de 5 preguntas en la lista, finaliza la ejecución.

4.3. Calificación de respuestas

Con el fin de evaluar las respuestas de los estudiantes a las diferentes preguntas, se procedió a implementar un algoritmo de calificación de respuestas. Esta calificación se realiza asignando a cada respuesta un número entero entre 0 y 4, donde 0 representa una respuesta perfecta y 4 es el caso de más errores.

En primer lugar, se utilizó únicamente el algoritmo de distancia de Levenshtein ($Dist_{levenshtein}$), con el cual, dado un conjunto de respuestas posibles $R = r_1, r_2, r_3, ..., r_n$ y una respuesta del alumno r_{alumno} la calificación de una respuesta dada se calculaba como:

$$Calificación = \text{mín}(\min_{\forall r \in R}(Dist_{levenshtein}(r, r_{alumno})), 4)$$

Este algoritmo, resultó ser válido para evaluar qué tan bien se ajusta la respuesta del alumno a las respuestas esperadas (ya que el maestro puede definir varias respuestas correctas a una misma pregunta).

Sin embargo, este criterio no es suficiente para considerar una respuesta como correcta. Durante una de las reuniones con ANEP (sección 6.1), surgió el requerimiento de que para que una respuesta sea considerada correcta, alcanza con que la respuesta del alumno, esté contenida dentro de una de las opciones. Dado que el algoritmo mencionado no se adapta a este requisito, el mismo fue descartado.

Para lograr esta nueva forma de evaluación de la respuesta, se utilizó la distancia de *Levenshtein* pero a nivel de palabra (para contemplar las faltas de ortografías), y se introdujo el algoritmo *Bag of words* [31] para poder analizar si en la respuesta del alumno se encuentran al menos todas las palabras de alguna de las respuestas posibles.

A continuación se presenta el algoritmo implementado para cumplir con este requerimiento de evaluación:

```
# Sean texto1 y texto2 arreglos de palabras
def distancia_textos(texto1, texto2):
    distancia = 4
    largo_palabras_texto1 = largo_palabras(texto1)
    permutaciones_texto2 = permutaciones(texto2, largo_palabras_texto1)

    Para cada permutacion_palabras en permutaciones_texto2:
        distancia_permitida = 0
        for i =1 i<= largo_palabras(texto1) i++:
            distancia_permitida += d_levenshtein(
                texto1[i],
                permutacion_palabras[i]
            )
        distancia = Minimo(distancia_permitida, distancia)

    Retornar distancia

distancia_minima = 4
Para cada respuesta_posible en respuestas_posibles:
    distancia_posible = distancia_textos(
        respuesta_posible,
        respuesta_alumno
    )
    distancia_minima = Minimo(distancia_minima, distancia_posible)

Retornar min_distancia
```

A pesar de que este algoritmo es de orden alto, consigue obtener muy buenos resultados sin afectar el desempeño de la aplicación. Esto se debe a que los cálculos se realizan solamente cuando cambia una respuesta, sin realizar nuevamente el cálculo para todas las demás.

Por otra parte, cabe aclarar que cuanto más larga sea la respuesta, peor será el desempeño del algoritmo. En la práctica, esta pérdida de desempeño no se observa dado que las respuestas generalmente son cortas.

Con el fin de brindarle retroalimentación inmediata al estudiante, a medida que este escribe la respuesta, se le asigna un color al texto en representación de la calificación obtenida en ese instante. Los colores utilizados para cumplir con lo anterior, fueron los siguientes:

- Calificación 0: Verde.

- Calificación 1: Verde Claro.
- Calificación 2: Amarillo.
- Calificaciones 3 y 4: Negro.

Se optó por no utilizar el color rojo de forma de evitar indicar enfáticamente que una respuesta no es correcta como sugerencia de la contraparte de ANEP.

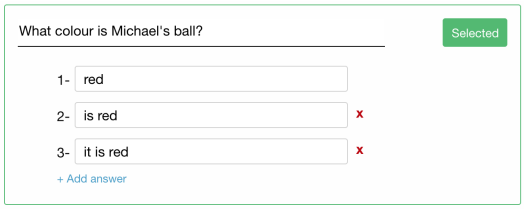

Figura 21: Calificación de respuesta

A modo de ejemplo, cuando el profesor genere la pregunta presentada por la Figura 21, el alumno podría responder a ella utilizando cualquiera de las oraciones indicadas: "*red*", "*is red*", "*it is red*". Además, como se considera correcto que la oración contenga estas palabras, el comportamiento esperado es el siguiente:

Respuesta	**Distancia**
it is blue	4
it is r	2
it is res	1
it is red	0
red is it	0
red	0
He has a red ball	0

4.4. Evaluación

En este capítulo se muestran los diferentes análisis realizados para evaluar las diferentes herramientas utilizadas y el generador de preguntas en términos de correctitud y tiempos de ejecución.

Debido a la poca cantidad de textos anotados para poder evaluar y con el fin de minimizar el sobreajuste[9] al corpus, muchas de las pruebas fueron realizadas partiendo de textos externos al corpus anotado en los que intuíamos que la aplicación podría dar resultados inesperados, para luego mejorar la aplicación y seguir iterando.

4.4.1. Herramientas de POS-tagging

Dado que la tarea de *POS tagging* debe realizarse para todos los tipos de regla durante la generación de preguntas y que existen diferentes herramientas para esta tarea, se procedió a evaluar cuál de estas ofrece mejores resultados en función del contexto.

Dentro del lenguaje de programación *Python* se destaca la librería de PLN ofrecida por Stanford [34], la cual es una de las más conocidas en el área y ha sido utilizada en proyectos anteriores obteniendo buenos resultados.

Por otro lado, también se decidió incorporar la librería *AllenNLP* a las pruebas para efectivamente comprobar cuál presenta mejores resultados y hacer un análisis objetivo acerca de cuál tendría un mejor impacto en el proyecto.

Dado que esta tarea es clave para el correcto funcionamiento y desempeño de la solución, se procedió a realizar tres tipos de evaluaciones:

1. Tiempo de ejecución de la librería a la hora de analizar una oración.
2. Correctitud de los resultados.
3. Evaluación en términos de desarrollo, la incorporación de la librería al proyecto.

Cabe destacar que para las pruebas se utilizaron textos que se consideran de niveles de inglés A1 y A2 (considerados básicos) y con 16 oraciones de largo promedio 55 palabras como datos de entrada. Por este motivo, los resultados pueden variar significativamente en caso de realizar un cambio en los datos de prueba, sobre todo en caso de que los textos tengan otra complejidad.

[9]El sobreajuste (*overfitting*), es el proceso de análisis de un determinado tema y obtener resultados enfocándose en un subconjunto particular de información, y no en el conjunto entero del problema. De esta forma, los resultados experimentales pueden ser muy buenos para el subconjunto seleccionado, pero no tanto para el conjunto en su totalidad

Herramienta	Desviación estándar	Tiempo Promedio de Ejecución
Stanford	0.584	0.821
Allen	0.951	1.341

Cuadro 8: Comparación de resultados de desviación estándar y tiempo promedio de ejecución en milisegundos en POS tagging para las herramientas Stanford y AllenNLP en milisegundos

Tanto *AllenNLP* como *Stanford* realizan el *POS Tagging* siguiendo la estructura planteada por el *Penn Treebank*, esto permite que se pueda hacer un análisis comparativo sin necesidad de modificar las salidas de ambas ejecuciones.

En cuanto al desempeño (en términos de tiempo de ejecución), los resultados en cuanto a la desviación estándar y la demora promedio se pueden apreciar en el Cuadro 8 respectivamente[10].

Como se puede apreciar en los resultados obtenidos, en promedio la librería *Stanford* mostró ser más rápida y con menor varianza, demorando en promedio casi la mitad del tiempo de ejecución de la librería *AllenNLP*.

La contundencia de estos datos demuestra que en caso de tener requerimientos específicos acerca del tiempo de demora, y en caso de trabajar con textos similares a los de las pruebas, se sugiere usar la librería de *Stanford*. Además, el tiempo de ejecución de la librería *AllenNLP* es más impredecible por ser su desviación estándar más alta.

En cuanto al análisis de correctitud, los resultados fueron los siguientes:

Herramienta	Etiquetados correctamente	Errores	Correctos/Totales
Stanford	879	6	0.993
Allen	873	12	0.986

Cuadro 9: Comparación de resultados de precisión en POS tagging para las herramientas Stanford y AllenNLP

[10] Para la ejecución de las pruebas de desempeño, se procedió a utilizar una MacBook Pro, con sistema operativo MacOS Mojave 10.14.3, un procesador 2.7 GHz Intel Core i5 y memoria ram de 8 GB 1867 MHz DDR3. Dada la incorporación de la herramienta Docker, estas pruebas pueden ser realizadas en casi cualquier plataforma, tendiendo a mejores resultados en máquinas Unix.

Ambas herramientas presentan muy buenos resultados en términos de correctitud del etiquetado, siendo la herramienta de *Stanford* sutilmente más precisa.

Finalmente, la elección de la librería *AllenNLP* para el proyecto se fundamenta principalmente por motivos de integración, investigación, y por el hecho de proveer todas las herramientas descriptas en el capítulo 2.2.2.

En particular, al momento de comenzar el proyecto, *Stanford* aún no contaba con una librería oficial para *Python* con todas las funcionalidades que se utilizan en este proyecto. Si bien sí existía una versión de *Stanford* para *Java* [35], se decidió no utilizar este lenguaje dado que no es el ideal para este tipo de proyectos (la mayoría de las herramientas de PLN en código abierto como *NLTK*, están implementadas en *Python*). Además, dado que dicha herramienta consume muchos recursos físicos, la integración con *Stanford* implicaba un riesgo de *performance* y de compatibilidad con los diferentes tipos de *hardware*.

No está de más destacar el hecho de que en los últimos meses *Stanford* sí ha desarrollado una librería en *Python* con las mismas funcionalidades que su versión en *Java*. Sin embargo, al momento de comenzar el proyecto no se contaba con versión estable, liviana y con suficiente documentación. Fue posible realizar estas pruebas una vez que la herramienta si se encontraba pública y con la documentación correspondiente [34].

4.4.2. Generador de preguntas

Para poder lograr un análisis objetivo de los resultados de la aplicación, no sólo es necesario analizar la usabilidad de la misma, sino también la correctitud de los ejercicios generados automáticamente.

En cuanto a la usabilidad, se procedió a hacer un análisis de los tiempos de respuesta del algoritmo de generación de preguntas en base a la totalidad de los textos del corpus (ver sección 4.5.1) con el objetivo principal de analizar que los tiempos de respuesta promedio no sean lo suficientemente grandes como para que resulte en una mala experiencia de usuario al docente.

Como se puede apreciar en el cuadro 10, los resultados respecto a los tiempos de respuesta para los textos de niveles A1 y A2 rondan los 12 segundos. Este resultado, si bien no es óptimo ya que influye en la usabilidad de la aplicación negativamente para los docentes, no es lo suficientemente significante como para

generar mayores impedimentos a la hora de utilizar la aplicación.

Oraciones	**Largo Prom.**	**Demora Prom.**	**Desviación estándar**
20	143.389	12.071	4.802

Cuadro 10: Resultados de ejecución del algoritmo de generación de preguntas en función del tiempo de ejecución. *Largo Prom.* representa el largo promedio en caracteres de cada oración. *Demora Prom.* es la demora promedio (en segundos) de la generación de preguntas para todas las oraciones. La *Desviación estándar* es la desviación estándar (en segundos) de la generación de preguntas para todas las oraciones

Si bien no se puede evaluar objetivamente la correctitud de la tarea de ordenar las preguntas (se parte de la base que el orden correcto es algo subjetivo), sí se puede medir la calidad de las preguntas generadas.

Para ello, se recurrió a utilizar el corpus anotado de preguntas y respuestas correspondientes a los niveles de inglés A1 y A2, que consiste en 20 textos de los cuales se desprenden 271 preguntas, explicado en detalle en la sección 4.5.1.

El corpus inicial de 20 textos fue separado en dos secciones, una para pruebas de desarrollo y otra para evaluación. El corpus de desarrollo consiste en 16 textos, 80 % del corpus inicial, mientras que el otro 20 % corresponde al corpus de evaluación utilizado solo a la hora de analizar los resultados finales de la aplicación. En la división del conjunto de textos se contempló el hecho que el conjunto de evaluación tenga la misma distribución que el conjunto de desarrollo, es decir, que no difieran sustancialmente dado que esto podría llevar a obtener malos resultados.

Como se puede apreciar en la tabla 11, es más común generar preguntas de tipo *what* o *who* que *where*, *when* o *what color*. Es por esto que si bien los resultados en las preguntas *what color* son buenos, no fue posible generar un corpus suficientemente grande como para tener una variedad de casos considerables en los cuales poder analizar el desempeño del algoritmo para la generación de este tipo de preguntas.

En lo que respecta a la evaluación de las respuestas, como el objetivo desde un principio fue encontrar una respuesta posible, se procedió a generar un corpus con respuestas posibles para cada una de las preguntas, considerando como correcta la generación cuando la respuesta generada se encuentra en las respuestas posibles

del corpus. Estos resultados se pueden apreciar en la tabla 12, donde los intentos son la cantidad de veces que se analiza si la respuesta es correcta o no, que coincide con la cantidad de preguntas generadas correctamente ya que solo es posible saber si una respuesta es correcta si la pregunta es correcta.

En cuanto a precisión, esta se calcula como la cantidad de respuestas generadas que se encuentran dentro de las preguntas generadas correctamente y que a su vez se encuentran en las respuestas del corpus para la pregunta correspondiente dividido la cantidad de preguntas generadas correctamente.

Tipo	**En corpus**	**Generadas**	**Incorrectas**	**Precisión**	**Recall**
Who	47	48	8	0.833	0.851
Where	4	2	1	0.500	0.251
What	33	34	6	0.824	0.849
When	1	2	1	0.500	1.000
What colour	1	1	0	1.000	1.000

Cuadro 11: Resultados de ejecución del algoritmo de generación de preguntas en función de la calidad de las preguntas. *Tipo* representa el tipo de pregunta. *En corpus* es la cantidad de preguntas en el corpus de validación para un tipo dado.

Tipo	**Respuestas generadas**	**Precisión**
Who	40	0.725
Where	1	1.000
What	28	0.600
When	1	1
What colour	1	1.000

Cuadro 12: Resultados de ejecución del algoritmo de generación de preguntas y respuestas en función de la calidad de las respuestas. *Tipo* representa el tipo de pregunta. *Respuestas generadas* es la cantidad de preguntas generadas correctamente, es decir, que se encuentran en el corpus. *Precisión* es la cantidad de respuestas generadas correctamente que se encuentran en las respuestas posibles para la pregunta correspondiente en el corpus.

4.5. Recursos generados

Tanto la identificación como generación de recursos han sido fundamentales no solo para la correcta construcción de los ejercicios, sino también para poder elaborar los diferentes análisis sobre el desempeño de las herramientas y del sistema generado.

La generación de recursos fue un desafío ya que no solo implicó la búsqueda de material que cumpliera con el nivel de inglés para el cual se construyó la aplicación, es decir, los requeridos por ANEP (niveles A1 y A2), sino que también sobre los recursos generados fueron necesarias varias correcciones para cumplir con los niveles de inglés mencionados, consultando a profesionales en el área.

Tanto los libros proporcionados por ANEP, como los recursos de la Universidad de Cambridge cumplieron un papel importante en la realización de estos recursos.

4.5.1. Corpus de oraciones junto a preguntas y respuestas generados a partir de estos

Para la evaluación de los diferentes ejercicios presentados en la sección 3.2 se decidió generar un conjunto de textos con preguntas y respuestas generadas a partir de estos. Para ello, se analizaron diferentes materiales educativos de Inglés, en particular para los niveles A1 y A2. Durante este análisis, se encontró un conjunto de textos de ESL[11] proporcionados por Earl Chang [32].

En base a 20 de estos textos, se anotaron las preguntas y respuestas consideradas apropiadas para los niveles anteriormente mencionados.

En total se generaron 271 preguntas para el conjunto de los 20 textos, las cuales fueron examinadas por un profesional en la lengua.

4.5.2. Corpus de oraciones de niveles A1 y A2 con su correspondiente pos-tagging

Para la evaluación de las diferentes herramientas para el *POS-tagging*, se etiquetaron manualmente 15 de los textos proporcionados por Earl Chang [32], utilizando las etiquetas definidas por el Penn Treebank.

[11] English as a Second Language

La lista completa de *tags* posibles se encuentra en la tabla 9. La elección de este conjunto de etiquetas se debe a que tanto la librería *Stanford* como *AllenNLP* las utilizan y dada la influencia que tienen ambas herramientas en el proyecto, pareció prudente seguir el mismo estándar. Este etiquetado también fue supervisado por un profesional de la lengua.

5. Aplicación y Arquitectura

5.1. Aplicación

Como se menciona anteriormente, la aplicación cuenta con dos secciones específicas para docentes y una creada para el uso de los alumnos a la que también pueden acceder los docentes. En este capítulo se detallan las tres secciones.

5.1.1. Sección docentes

En esta sección, los docentes pueden ingresar un texto en inglés con el fin de generar preguntas y respuestas, para luego formar un ejercicio.

Debido a que dicha generación puede contener errores, requiere de la supervisión docente para validar las preguntas y respuestas antes de crear el ejercicio. Por este motivo, la sección se encuentra protegida mediante una contraseña de uso docente, con el fin de que los alumnos no generen sus propios ejercicios. Esta funcionalidad puede apreciarse en la figura 22.

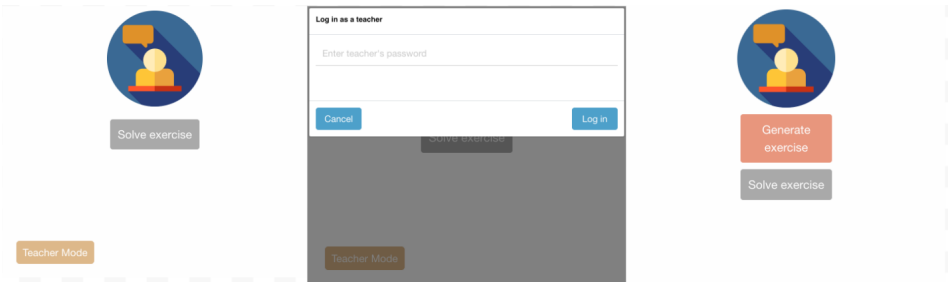

Figura 22: Procedimiento de acceso a la interfaz docentes

Una vez que el docente proporciona la contraseña y accede a la interfaz docentes, se presenta un espacio para introducir el texto en el cual desea basar el ejercicio, como se puede apreciar en la figura 23. El texto introducido estará también disponible para los alumnos al momento de resolver el ejercicio.

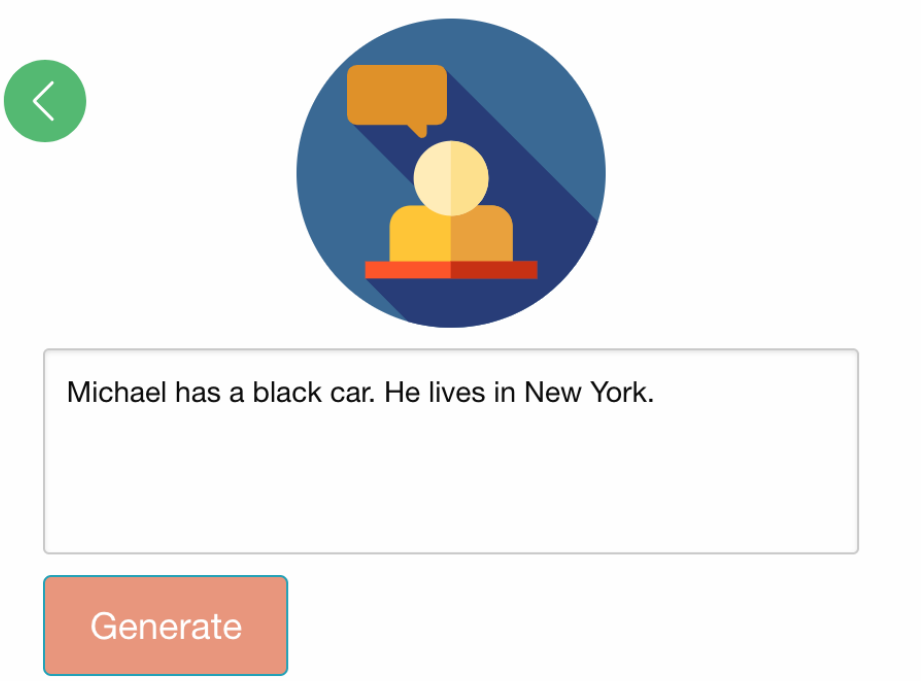

Figura 23: Ingreso del texto para la generación del ejercicio

Una vez que el docente selecciona la opción *Generate*, la aplicación procesará el texto para luego desplegar un conjunto de preguntas y respuestas auto generadas (ver figura 24)

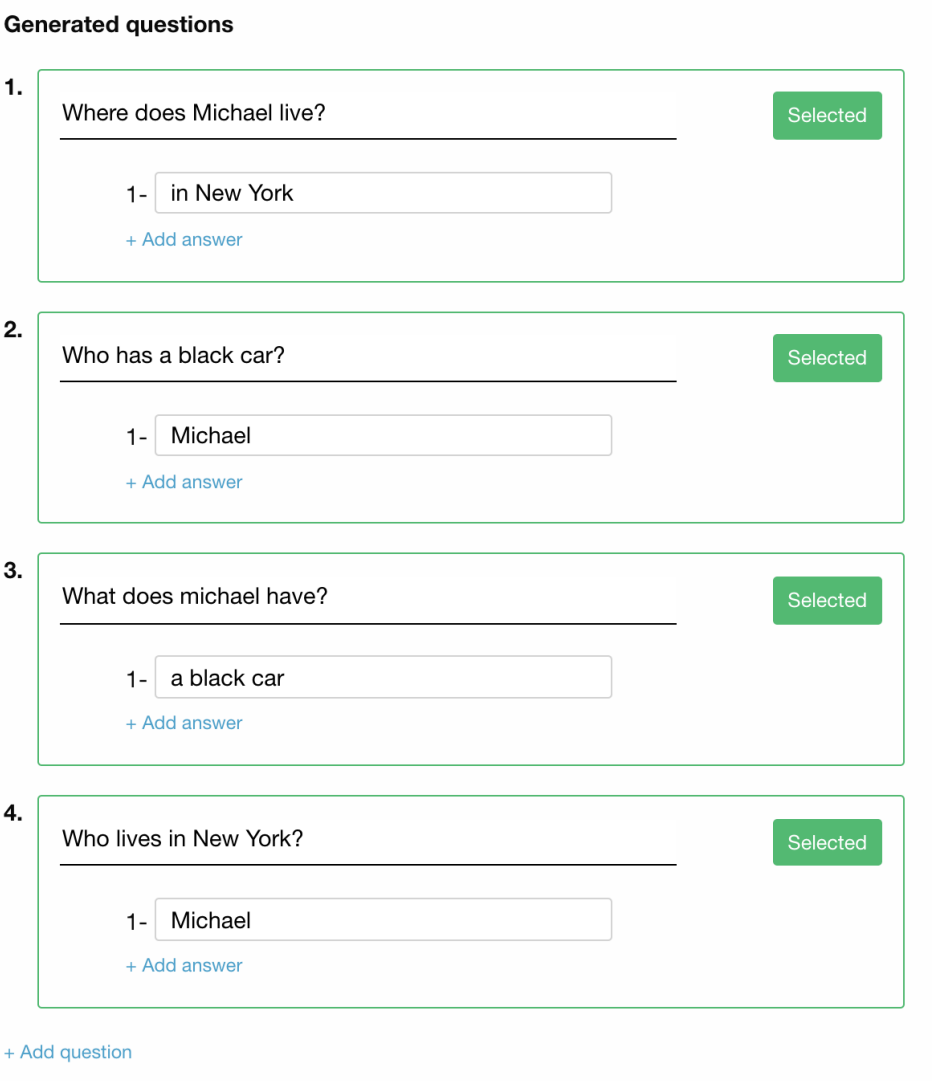

Figura 24: Visualización de las preguntas auto generadas en la sección Alumnos

Como se puede apreciar en la figura 24, el docente puede realizar varias acciones, entre ellas:

- Agregar una nueva pregunta
- Editar una pregunta
- Agregar una respuesta a una pregunta

- Editar una respuesta de una pregunta
- Seleccionar qué preguntas estarán disponibles para los alumnos

Estas funcionalidades son importantes, ya que le ofrecen al docente la posibilidad de corregir posibles errores en el contenido generado. Además de errores en cuanto a sintaxis o semántica, se ofrece la posibilidad de evaluar las preguntas generadas en términos de complejidad, permitiendo realizar modificaciones o descartar cualquier contenido del ejercicio.

Por último, como se aprecia en la figura 25, los docentes deben asignarle un nombre al ejercicio para poder guardarlo.

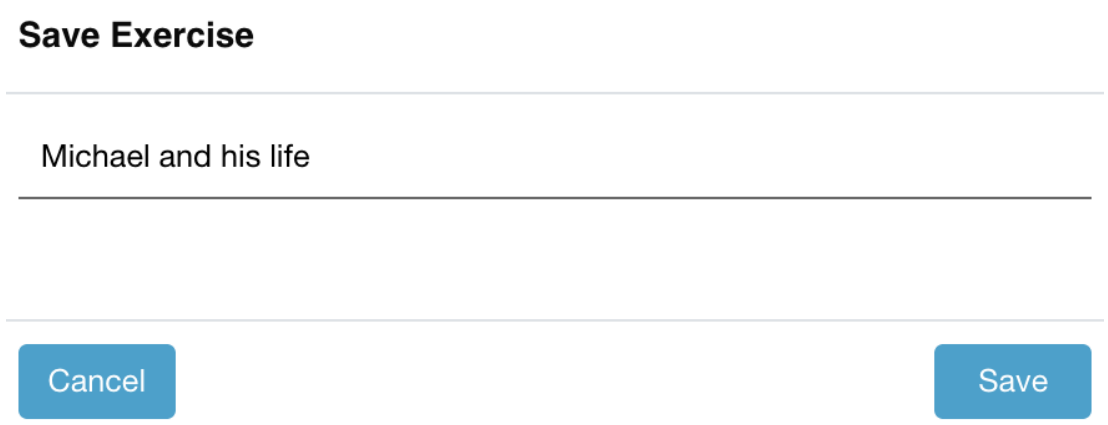

Figura 25: Ingresar nombre del ejercicio

5.1.2. Sección alumnos

Esta sección es la que utilizan los alumnos al acceder a la aplicación. Los alumnos sólo tienen la opción de ingresar a esta sección ya que carecen de la contraseña para poder ingresar a las demás secciones.

Como se aprecia en la Figura 26, aquí se despliega una lista de todos los ejercicios disponibles para que los alumnos puedan resolver.

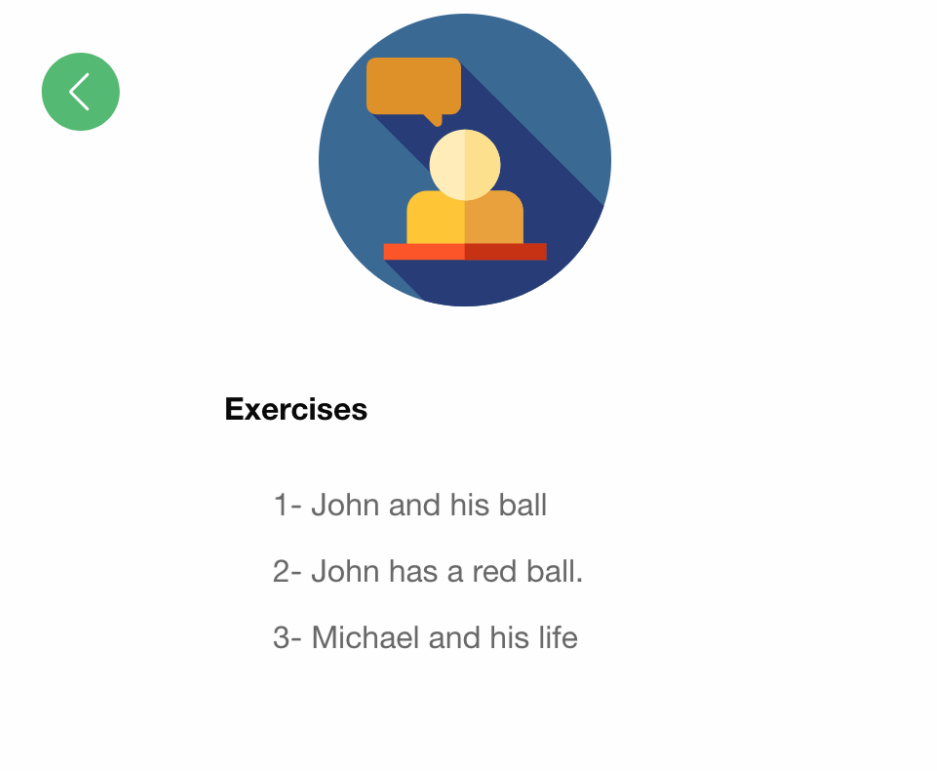

Figura 26: Selección del ejercicio

Una vez que el alumno selecciona un ejercicio, se despliega el texto del ejercicio junto a la lista de preguntas a resolver, seguidas de un espacio para que el alumno pueda escribir su respuesta (ver figura 27).

Como se puede apreciar en la imagen, cada respuesta presenta un color diferente en función de qué tan cercana está a las posibles respuestas correctas. Esta funcionalidad se implementa aplicando el algoritmo de asignación de colores a las respuestas, explicado en el capítulo 4.3.

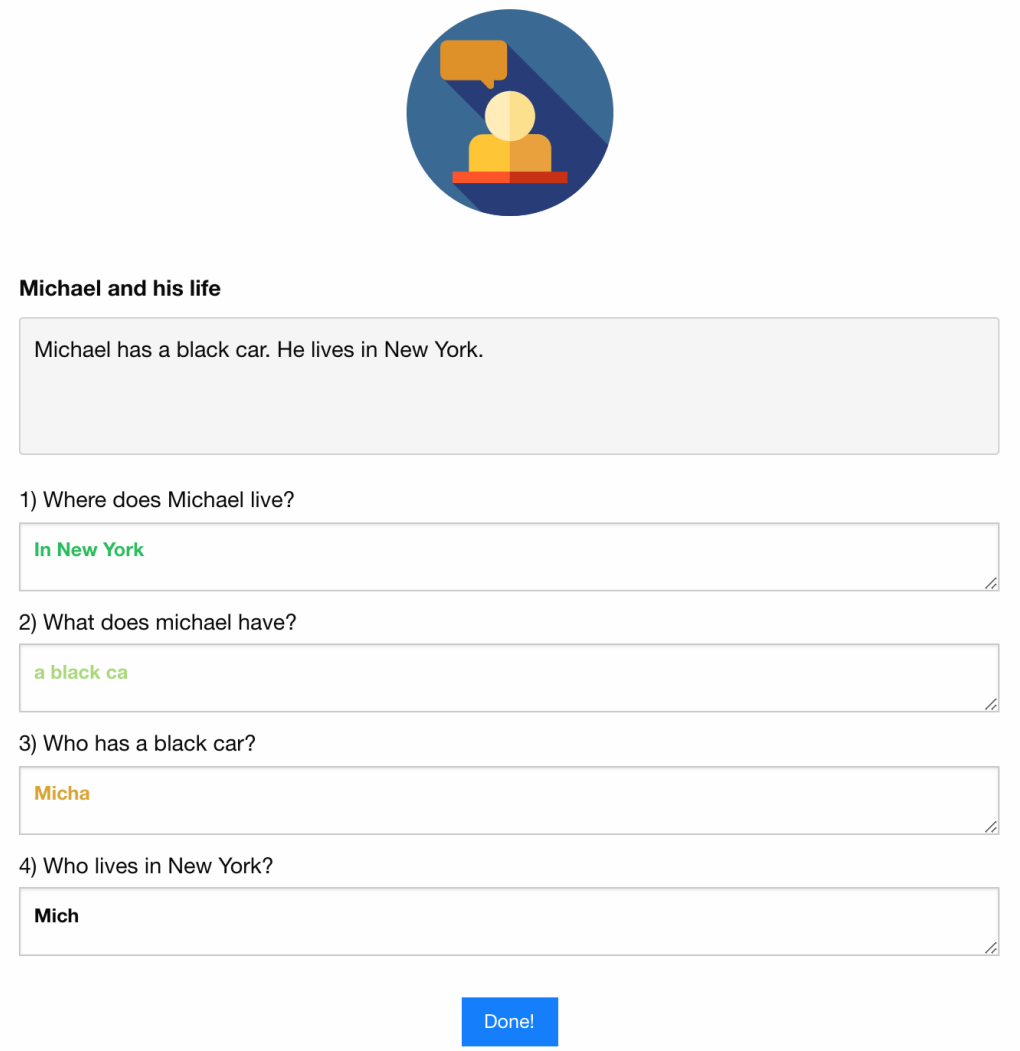

Figura 27: Resolución del ejercicio

Finalmente, como se aprecia en la figura 28, cuando el alumno considera por terminado el ejercicio, se despliega una cartel indicando la cantidad de respuestas correctas[12].

Luego de ver los resultados, el estudiante puede decidir si volver al menú principal a elegir otro ejercicio, así como también puede decidir iterar sobre sus respuestas y así obtener una mejor calificación.

[12]Es en este momento que las respuestas del alumno se envían al *backend* para ser almacenadas.

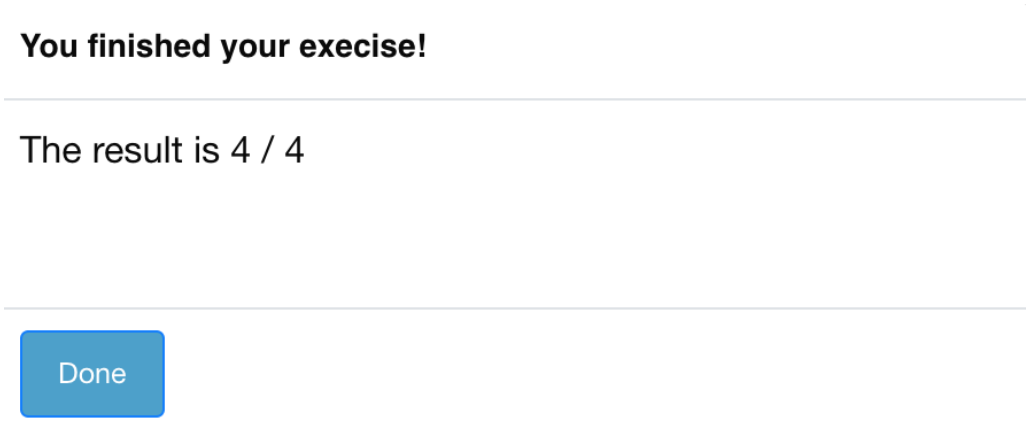

Figura 28: Presentación de los resultados del ejercicio

5.1.3. Sección estadísticas

Esta sección abarca todo lo que refiere a los datos obtenidos de ejercicios generados y las resoluciones de los alumnos sobre estos. El objetivo es poder brindar una interfaz en la cual se pueda visualizar de forma clara todos los resultados de la interacción con el programa. Entre los datos que se presentan, se destacan los ejercicios generados, ediciones de ejercicios y resoluciones por parte de los alumnos.

Esta parte es accesible mediante una URL diferente a la de las dos secciones anteriores, y está protegida por usuario y contraseña ya que la visualización y edición de los datos debe estar restringida.

Potencialmente, la información registrada puede ser de mucha utilidad ya que los docentes podrían acceder a las resoluciones de los alumnos a los diferentes ejercicios propuestos e identificar fortalezas y debilidades.

Por otra parte, con fines de investigación, esta sección presenta información respecto a qué tan bien se realiza la generación de preguntas, pudiendo visualizar qué tantos cambios necesitó realizar el docente para llegar al ejercicio final.

Como se puede apreciar en la figura 29, en esta sección se presentan tres opciones:

- *Exercise questions*: visualización por preguntas
- *Exercises*: visualización por ejercicios

- *Resolutions*: visualización por resoluciones de los alumnos

Figura 29: Menú de la sección de estadísticas

Visualización de estadísticas por preguntas

Este tipo de visualización tiene como objetivo presentar las preguntas de todos los ejercicios (ver figura 30), y los datos asociados a las mismas.

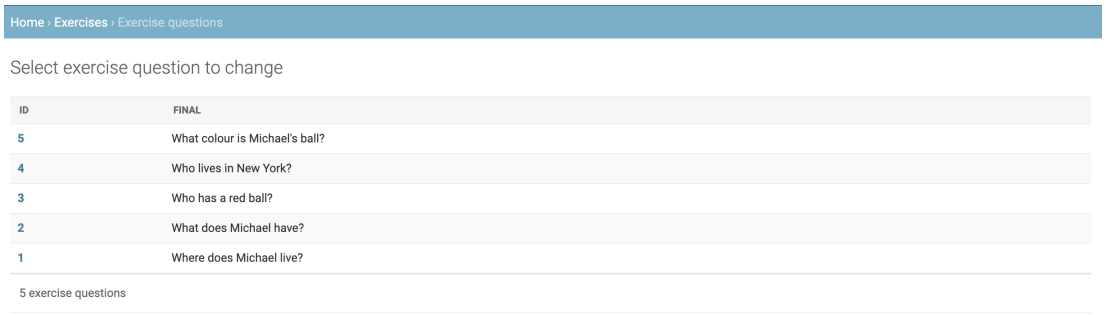

Figura 30: Menú de preguntas. Sección estadísticas.

Como se aprecia en la figura 31, se puede visualizar a qué ejercicio pertenece cada pregunta (por lo tanto ver cual fue el texto que ingresó el docente para generarla), cual fue la pregunta generada por el sistema (*initial*) y cual fue la versión final de la pregunta luego de las correcciones del docente (*final*).

Con respecto a las respuestas correctas para la pregunta generada, se muestra en la primera fila la respuesta generada por el sistema (*initial*) y la versión final luego de las correcciones del docente (*final*). En el resto de las filas se muestran todas aquellas respuestas agregadas por el docente (en caso de haber).

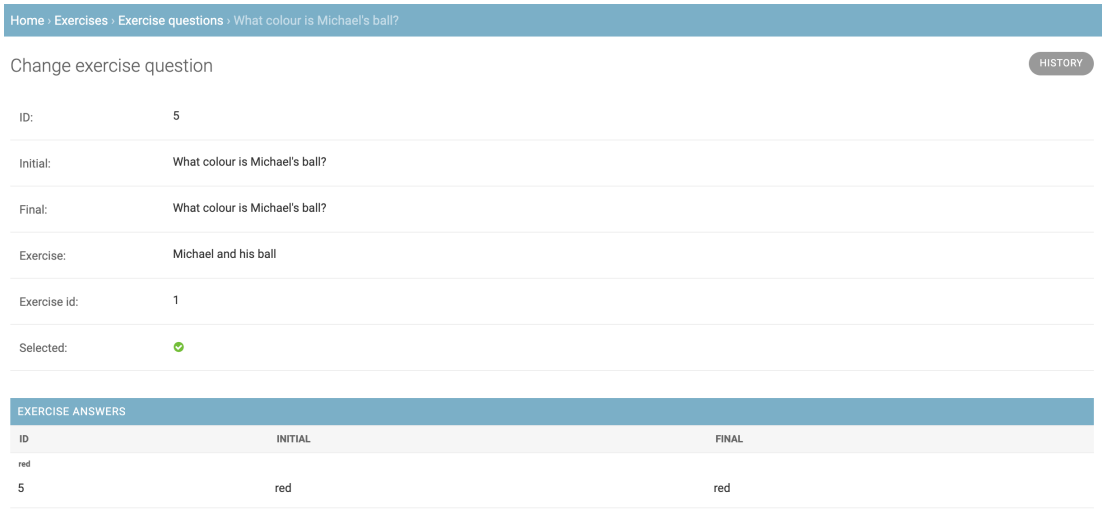

Figura 31: Sección estadísticas para una pregunta

Visualización de estadísticas por ejercicios

Esta pantalla tiene como objetivo poder brindar un panorama general de la información obtenida en la generación de un ejercicio por parte de los docentes.

Como se puede apreciar en la figura 32, inicialmente se presenta el nombre del ejercicio (*name*) seguido por el texto utilizado por el docente para generar el mismo (*text*).

Luego se listan todas las preguntas que fueron auto generadas para dicho ejercicio. En la primer columna y bajo el campo *initial*, se muestra la versión generada automáticamente y sin modificaciones (no añadida de forma manual) de cada pregunta. Bajo el campo *final*, se presenta la versión final de la pregunta, luego de las correcciones de los docentes. Finalmente también si indica si la pregunta fue utilizada (*selected*) para conformar el ejercicio.

En caso de querer acceder a los detalles de cada pregunta, es decir, a la sección , basta con acceder al *id* de la misma.

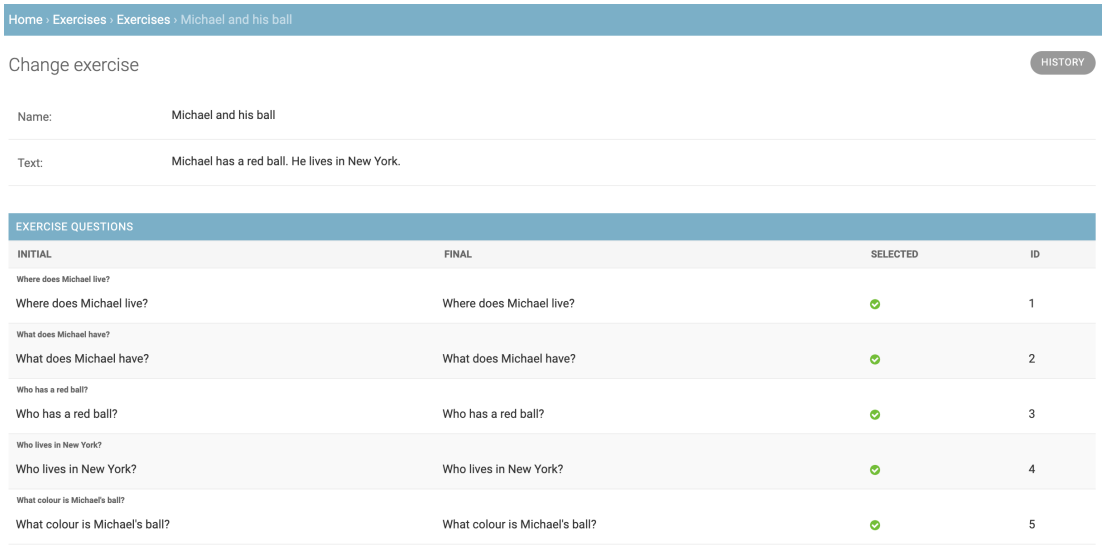

Figura 32: Sección estadísticas para un ejercicio

Visualización de estadísticas por Resoluciones de los Alumnos
En esta última visualización de datos, se presentan las resoluciones de los alumnos para cada uno de los ejercicios (ver figura 33).

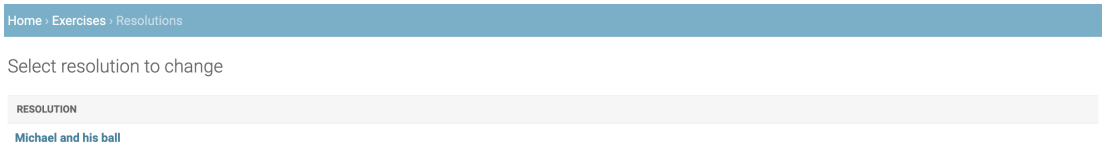

Figura 33: Visualización por resoluciones de los alumnos

Como se aprecia en la figura 34, en esta pantalla se muestra el nombre del ejercicio (*exercise*), y luego se proporciona una lista con el conjunto de preguntas del ejercicio junto a la respuesta brindada por el alumno.

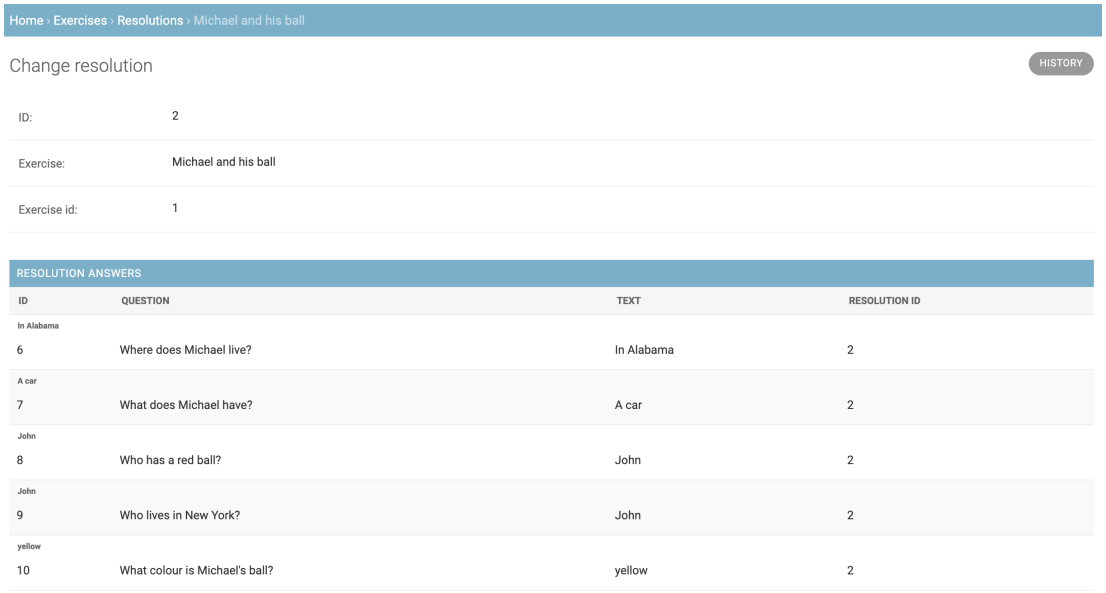

Figura 34: Sección estadísticas para una resolución

5.2. Arquitectura

Dadas las características del proyecto, era necesario contar con un sistema que pueda soportar los requerimientos de:

- Soportar gran cantidad de usuarios
- Tener soporte de almacenamiento
- Tener alta disponibilidad
- Funcionar a una velocidad razonable

Para solucionar esto es que se decidió trabajar con una arquitectura cliente servidor, y concentrar los costos computacionales en el servidor, de forma que las ceibalitas, a la hora de utilizar este sistema no tengan problemas de desempeño.

5.2.1. Estructura general

En el sistema se pueden identificar 2 entidades principales. Por un lado el Cliente, el cual se ejecuta en las computadoras personales de los estudiantes (ceibalitas).

Actualmente, se encuentra configurado de forma tal que todos los archivos para su ejecución se obtienen a partir de un *Bucket* S3 de AWS [33] mediante pedidos HTTP, como se muestra en la Figura 35.

Por otra parte, se puede identificar el servidor (actualmente configurado en una computadora provista por la Facultad de Ingeniería), donde podemos distinguir por un lado el módulo que se encarga de toda la lógica de la generación de preguntas, y por otro la Base de Datos.

Con el fin de utilizar la infraestructura provista por la Facultad de Ingeniería para correr el Servidor, se presentó el requerimiento de que este solamente utilice el puerto 80 del equipo provisto para exponer sus servicios. Por este motivo se decidió tener la lógica junto a la base de datos, donde la única entidad que expone su puerto a la web es la API implementada para la lógica.

Por otro lado, todos los archivos del *Frontend* (Web) como archivos HTML, CSS y Javascript fueron alojados en un servidor de archivos estáticos de AWS (S3 Bucket de AWS). Este esquema puede visualizarse en la figura 35

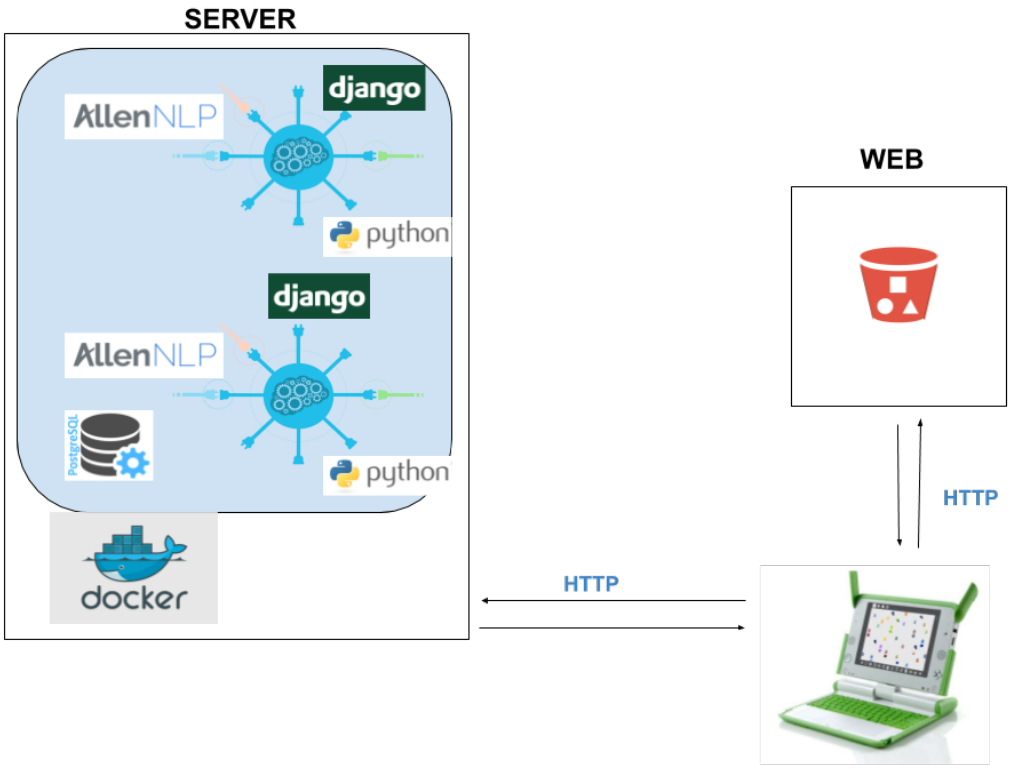

Figura 35: Esquema general de la arquitectura

5.2.2. Herramientas Utilizadas

Dada la gran cantidad de servicios y la necesidad de que el sistema pueda ser transferido a otros servidores con facilidad (portabilidad), es que se decidió utilizar la herramienta Docker [37].

Esta herramienta provee una capa de abstracción entre los diferentes recursos, como puede ser la base de datos, la aplicación *Python (Backend)* y la web (*Frontend*). De esta forma y utilizando los beneficios que ofrece esta herramienta, alcanza con ejecutar un solo comando para tener aplicación corriendo, independientemente del hardware subyacente [13]. Esto permitió facilitar la configuración durante la etapa de desarrollo, abstrayendo las características físicas en las cuales

[13] En algunos casos, como puede ser cuando se utiliza *proxy*, si es necesario realizar configuración adicional

se corrió la aplicación.

Por otra parte, la herramienta *Docker* permite indicar en cuántos nodos (o instancias de la aplicación) correrán cada uno de los recursos mediante un parámetro, es decir, en caso de que la cantidad de alumnos sea muy alta, puede configurarse para que la cantidad de nodos que corra la aplicación sea más alta. Los nodos se dividen en:

- Base de datos
- API
- Web

Por otra parte, se utilizó el lenguaje de programación *Python* para la interacción entre la API, la base de datos y las diferentes librerías. La elección de dicho lenguaje viene dada por la importancia y distribución de dicho lenguaje en el área del PLN.

Por otra parte, para la construcción de la API en sí, se recurrió a hacer uso de *Django* [38], un *Framework* de *Python* dedicado a la implementación de aplicaciones web, con soporte HTTP como protocolo de transferencia. La elección de este *Framework* se fundamenta por el hecho de que ha probado escalar para soportar el manejo de múltiples usuarios concurrentes y aplicaciones de gran porte. Además, ha sido utilizado previamente por los integrantes del equipo, por lo que su integración resultó práctica y sencilla durante el proceso de desarrollo.

En términos de persistencia, se decidió utilizar *PostgresSQL*, dado que al igual que *Django*, está diseñada para escalar y servir a múltiples pedidos a la vez de forma concurrente. Además, el *Framework* utilizado ofrece soporte para dicha base de datos.

En cuanto a la construcción de página web, se utilizó la librería *ReactJS* [39], que es uno de los *Frameworks* más populares para el desarrollo web en la actualidad.

5.2.3. Desempeño

Uno de los aspectos destacados de la arquitectura, es que está diseñada para escalar rápidamente para brindar servicio a múltiples usuarios.

A su vez, dadas las características de *ReactJS* como SPA[14] (Single Page App), donde el navagador descarga mediante un pedido HTTP los archivos del sitio una única vez, se presenta una ventaja en cuanto a $performance$ por parte de la aplicación.

Cabe destacar que luego de que se carga el sitio por primera vez, los únicos pedidos que se realizan son hacia la API del *Backend* por datos específicos. Por ejemplo, durante el flujo completo de uso de la aplicación, es decir, entrar a la página, elegir un ejercicio y resolverlo implica un total de 6 $requests$, totalizando unos 2.8 MB. Por lo que aún con internet limitada, el programa funciona de forma correcta.

Otro característica importante, es el hecho de que la base de datos es compartida para todos los docentes, por lo que ellos pueden acceder a los ejercicios de otros docentes. El hecho de poder compartir el conocimiento (en este caso el conocimiento de un buen ejercicio) se considera fundamental para mejorar a la hora de ofrecer nuevos recursos a los alumnos. De esta forma los docentes pueden obtener métricas acerca de cómo responden los alumnos a un mismo ejercicio e interactuar con otros entes de la educación para poder generar material educativo, público y de calidad.

[14] Una página web SPA es una página que interactúa con el navegador cambiando dinámicamente la página en base a la interacción con el servidor. A diferencia de las páginas web comunes que cargan páginas enteras en función de la interacción con el usuario. El objetivo principal es tener transiciones más fluidas para poder hacer que el sitio parezca más una aplicación nativa reduciendo la cantidad de pedidos y por ende los tiempos de espera. En este tipo de páginas web, los archivos HTML, *JavaScript* y CSS necesarios se descargan en un único *request*[36].

6. Actividades de vinculación

Durante el proyecto se realizaron diferentes actividades de vinculación, con el fin de validar los requerimientos relevados así como las funcionalidades desarrolladas hasta el momento.

En primer lugar, se trabajó directamente con una contraparte de ANEP de modo de lograr retroalimentación directa durante todo el proyecto, con la cual se fijaron dos reuniones presenciales. Por otro lado, también se realizó una presentación en el Foro de Lenguas de ANEP y una visita a una escuela, con el objetivo de compartir el trabajo y recibir más opiniones y sugerencias en cuanto al funcionamiento de la aplicación.

6.1. Reuniones con ANEP

Desde el inicio del proyecto se estableció una comunicación constante con el encargado de Políticas Lingüísticas de ANEP: Aldo Rodríguez.

La reunión inicial tuvo lugar el día 2 de Mayo de 2019, en la antigua sede de Políticas Lingüísticas ubicada sobre la calle 18 de Julio (Montevideo).

En esta primera reunión, Aldo presentó sus inquietudes acerca de algunos aspectos a mejorar, como por ejemplo la escasez de material de práctica para los estudiantes de diferentes niveles, en particular de material que hiciera uso de las herramientas informáticas que los alumnos tienen, más específicamente, de la ceibalita.

Se planteó la posibilidad de la creación de un programa de generación automática de preguntas y respuestas en base a textos en inglés, teniendo en cuenta las diferentes restricciones del dominio en el cual los alumnos deben desarrollar sus habilidades.

Entre los diferentes aportes de Aldo, destacó el hecho de que los ejercicios de esta índole contemplan dos tipos de preguntas: aquellas cuya respuesta se encuentra explícita en el texto, y aquellas donde existe la necesidad de comprender semánticamente el texto para proveer una respuesta correcta.

Otro de los aspectos que parecieron pertinentes a tener en cuenta era el hecho de que los ejercicios debían ser acordes a los niveles de Inglés Movers o Starters en adelante, por lo que consideramos necesario soportar preguntas del estilo *What*, *What colour is*, *Who*, *When* y *Where*.

Por último, como funcionalidad adicional para la aplicación, surgió la idea de permitir que el docente pueda proponer varias respuestas correctas a una misma pregunta, dado que generalmente las preguntas pueden ser respondidas de manera diferente. Al finalizar esta reunión, se nos brindó una buena cantidad de material educativo de inglés para tener un mayor entendimiento acerca del nivel y la forma de educación del mismo.

Luego de estar en una etapa avanzada del desarrollo de la aplicación, se procedió a realizar una segunda instancia de encuentro el día 12 de Noviembre de 2019, en la nueva sede de Políticas Lingüísticas sobre la calle Agraciada (Montevideo).

En esta reunión, Aldo pudo ver la aplicación en su totalidad, donde dio a conocer algunos aspectos de mejora, sobre todo en el área de las respuestas. Explicó que en el enfoque actual de la educación de las segundas lenguas en Uruguay, lo más importante es lograr la comunicación, más allá de los posibles errores de ortografía. Además, mencionó que las respuestas a las preguntas pueden no estar completas, o tener sus palabras en el orden equivocado, pero aún así considerarse correctas dado que el alumno consiguió comprender el aspecto general de la respuesta (ver capítulo 4.3).

Más allá del buen intercambio de ideas y la satisfacción en términos generales de ANEP respecto al programa, se presentó la idea de poder validar los ejercicios en alguna escuela del país, en particular una que estuviera en el programa *Inglés Sin Límites*. Aldo nos dejó en contacto con la directora de una escuela rural que luego visitamos y en la cual presentamos el proyecto.

6.2. Foro de lenguas

El 12 de Octubre de 2019 tuvo lugar el Foro de Lenguas, donde se brinda un espacio para realizar presentaciones y compartir conocimiento en el área de la lengua.

En particular, se dio la oportunidad de presentar el trabajo (ver figura 36) frente a un grupo de maestros interesados en el área. El objetivo fue el de compartir los avances de la aplicación en general (figura 37), de forma de dar a conocer el trabajo realizado y de intercambiar opiniones con los presentes respecto al trabajo.

Figura 36: Presentación en foro de lenguas, aspectos generales

La presentación duró aproximadamente unos 30 minutos, durante los cuales se exhibieron los aspectos teóricos detrás de la aplicación, para luego entrar en detalle en las diferentes funcionalidades que ofrece.

Finalmente, hubo un espacio para intercambiar ideas, donde los presentes dieron a conocer sus puntos de vista y realizaron comentarios respecto a la aplicación. Una de las inquietudes comunes que surgió entre los maestros fue el hecho de conocer qué rol ocupa el docente en la generación del ejercicio. Se aclaró que no solo implica la elección del texto, sino que también debe estar presente en la elección de preguntas y respuestas (dado que el generador puede tener errores).

Debido a la cantidad de maestros provenientes de diferentes escuelas de todo el país, esta instancia también sirvió para conocer diferentes percepciones respecto al uso de la tecnología en primaria. En particular, también contribuyó para fortalecer el lazo entre la Universidad y la Educación Pública a nivel de primaria.

En general, los comentarios recibidos fueron muy buenos y de utilidad para continuar desarrollando y mejorando la aplicación. Consideramos que la participación realizada en el foro fue sumamente positiva, porque además de la retroalimentación recibida, pudimos notar el entusiasmo y ganas de los maestros de utilizar la aplicación implementada.

Figura 37: Presentación en foro de lenguas, aplicación

6.3. Visita a escuela

El día 3 de Diciembre de 2019 visitamos la Escuela rural Número 27 de Canelones, situada en la ruta 33, km 49.200. En esta oportunidad fuimos acompañados por los supervisores del proyecto Aiala y Luis, así como también por Alejandro Tosi, quien presentaría también su proyecto "Aplicaciones Lúdicas de Soporte a la Enseñanza de Lengua" [4] a los alumnos de la escuela.

Fuimos recibidos por la maestra directora, Martha Gómez, quien fue muy amable en dejarnos participar de esta instancia en su escuela, con el fin de compartir nuestro trabajo.

Una vez que se nos dio la bienvenida, fuimos presentados frente a la clase correspondiente a los niveles de 4^{to}, 5^{to} y 6^{to} año, de aproximadamente 11 alumnos.

En primer lugar se procedió a escuchar a la maestra y a los alumnos respecto a las dificultades que enfrentan en el día a día con el aprendizaje de Inglés. Entre los comentarios más importantes, se destaca el hecho de que los alumnos realizan un uso frecuente de las herramientas informáticas, en particular de la Ceibalita. También se mencionó el hecho de que gran parte del trabajo se realiza con las Ceibalitas y que sin dudas son de gran apoyo para los maestros y alumnos.

Luego de las presentaciones y de intercambiar opiniones, Alejandro Tosi procedió a presentar e instalar las aplicaciones lúdicas de su proyecto (ver 2.3.6) a los

alumnos. Al instalar estos ejercicios en sus respectivas Ceibalitas, los alumnos no dudaron en empezar a investigar y trabajar con las herramientas. Se los notó entusiasmados de tener estos juegos a su alcance y motivados de saber que estaban aprendiendo Inglés mientras jugaban y utilizaban la computadora.

Durante la actividad realizada por Alejandro, tuvimos un espacio para poder conversar con la directora acerca de nuestro proyecto (figura 38).

Figura 38: Presentación de la aplicación a la directora de la Escuela

En primer lugar, realizamos una breve introducción a nuestro trabajo, particularmente nuestros objetivos y avances hasta el momento, para luego finalizar con una pequeña demostración del funcionamiento de la aplicación.

La reacción de Martha fue positiva y se la notó interesada, principalmente en el hecho de que como maestra pudiera apoyarse en la aplicación para la generación de ejercicios para los alumnos. Además, destacó como importante la posibilidad de decidir qué preguntas y respuestas generadas deberían conformar un ejercicio.

Por otra parte, reconoció que otro de los aspectos interesantes era que la aplicación pudiera corregir los ejercicios automáticamente con colores, de forma que los alumnos obtengan retroalimentación rápida y sencilla.

Finalmente, destacó la importancia de visualizar las respuestas de los alumnos para cada uno de los ejercicios, así como también la posibilidad de poder compartir

ejercicios.

A pesar de que también nos hubiera gustado utilizar la aplicación con los alumnos, al momento de la visita no contábamos con una versión terminada como para trabajar con ellos. Si bien uno de los objetivos para este año era el de visitar una escuela para usar la aplicación con los alumnos, este fue descartado debido a la situación de emergencia sanitaria que atraviesa el país[15].

Una vez finalizada la jornada, nos despedimos de los alumnos y maestros, quienes nos agradecieron por la visita. Se mostraron a la espera de empezar a utilizar todas las herramientas de forma continua y en colaborar con los proyectos.

Consideramos que esta instancia fue positiva en muchos sentidos. En primer lugar, por el hecho de haber evaluado con éxito la aplicación con la maestra directora, quien forma parte del grupo de usuarios a quienes está destinada. Pero principalmente por la interacción con los alumnos, quienes nos brindaron todo su entusiasmo y alegría para trabajar, lo cual fue una gran motivación para continuar trabajando y mejorando lo realizado hasta el momento.

[15]Debido a la pandemia causada por el virus COVID-19, el día 13 de Marzo de 2020, se declaró en estado de emergencia sanitaria a todo el territorio uruguayo. Entre otras medidas, esto implicó la suspensión de las clases presenciales en todas las escuelas del País.

7. Conclusiones y Trabajo a Futuro

En esta sección se presentan las conclusiones obtenidas durante la realización del proyecto en cuestión. Además, se presentan diferentes propuestas de trabajo a futuro, con el fin de continuar mejorando lo realizado.

7.1. Conclusiones

De acuerdo a los objetivos planteados al inicio, se logró construir una aplicación en línea que ofrezca la generación automática de ejercicios. En particular ejercicios de preguntas y respuestas en inglés, de forma de contribuir con la educación de esta segunda lengua.

La aplicación fue elaborada cumpliendo con todos los requisitos planteados al inicio del proyecto, por lo que consideramos satisfactoria su implementación. Además, debido a las características de la arquitectura mencionadas en el capítulo 5.2, la misma estará disponible para ser configurada en una infraestructura diferente a la actual y para utilizarse en años siguientes.

En lo que respecta a los recursos generados, consideramos que sin dudas uno de los mayores desafíos fue la generación del corpus de preguntas y respuestas anotado. Si bien son preguntas relativamente fáciles y a partir de textos considerados como sencillos, el costo de validar que sean correctas y que efectivamente cumplan con los niveles adecuados de inglés es alto. Además de la generación de preguntas y respuestas, fue necesario realizar la anotación gramatical (*POS tagging*) de las preguntas para evaluación de las diferentes herramientas.

Sin dudas haber contado con un corpus desde el inicio, hubiese facilitado la generación de preguntas. Sin embargo, la realización de este corpus permite ofrecer un recurso nuevo que puede resultar de utilidad para trabajos posteriores y además puede ser extendido y mejorado.

Dado que durante el proyecto pudimos confirmar el esfuerzo que implica la generación de un corpus pequeño con fines de evaluación, fue correcta la decisión de no optar por el enfoque de aprendizaje automático para la generación de preguntas. Además, los resultados obtenidos se encuentran dentro de lo esperado, por lo que se considera que el enfoque de reglas utilizado fue el adecuado. De hecho, al momento de tomar la decisión ninguno de los enfoques mencionados en el capítulo 2.3 presentaba diferencias notorias en cuanto a sus resultados.

Uno de los objetivos planteados al comienzo del proyecto fue el de investigar nuevas herramientas para el uso de recursos y técnicas de PLN.

En particular, consideramos que fue clave la elección de la librería *AllenNLP* como herramienta principal en el desarrollo de la solución. Por los resultados obtenidos y las pruebas realizadas, creemos que es una herramienta a tener en cuenta para futuros trabajos en el área, sobre todo por la facilidad de su uso e instalación.

Por otra parte, la librería de *Stanford NLP* mostró en la mayoría de los casos mejores tiempos de procesamiento y precisión en los análisis de textos. Creemos que si la aplicación fuera construida utilizando esta herramienta para el procesamiento de texto, los resultados podrían ser aún mejores.

En general, consideramos que la solución realizada se alinea con los objetivos planteados al inicio del proyecto y contribuye en buena manera al problema de la falta de recursos para le enseñanza de inglés en las escuelas.

7.2. Trabajo a Futuro

De forma de continuar añadiendo funcionalidades y mejorando el presente trabajo, se presentan a continuación los diferentes aspectos que consideramos podrían ser realizados.

7.2.1. Registro de Usuarios

Actualmente se cuenta con una modalidad dedicada exclusivamente a docentes y otra para estudiantes, donde la sección docentes es accedida mediante una única contraseña común para todos. Debido a esta limitación, en la práctica los docentes pueden acceder a los ejercicios generados por los demás docentes y ver trabajos realizados por los alumnos.

Se propone realizar una sección dedicada al registro de docentes y alumnos, con el fin de que cada usuario cuente con su propio perfil al momento de utilizar la aplicación. Para el caso de los docentes, esto implica que los ejercicios serán accedidos solamente por el docente que los generó, con la posibilidad de indicar con qué alumnos desea compartir el ejercicio. En el caso de los alumnos, implica que tendrán acceso solamente a los ejercicios realizados por su docente.

En particular, también consideramos importante la posibilidad de que los maestros puedan compartir los ejercicios generados entre sí, con el fin de permitir el

trabajo colaborativo.

7.2.2. Nuevos tipos de Preguntas

Además de continuar evaluando y mejorando las preguntas que se pueden generar, se propone añadir nuevos tipos de pregunta al conjunto de ejercicios generados. En particular, se podrían añadir preguntas de la forma:

- *How many ... ?*
- *How old ... ?*
- *Which ...?*
- Con respuesta *Yes / No* (Yes or No questions).
- Con respuesta múltiple opción. En este caso se podría hacer uso de la resolución de entidades nombradas para extraer las entidades nombradas y utilizarlas como respuestas posibles cuando solo una es correcta.

7.2.3. Generación de Preguntas a partir de Imágenes

Durante el análisis de los tipos de preguntas y respuestas que se utilizan generalmente para los niveles básicos de inglés, se observó que muchos de los ejercicios se basan en la utilización de una imagen (por ejemplo la habitación de un niño) para realizar preguntar respecto a la misma.

Se propone implementar la generación de preguntas y respuestas a partir de una imagen, utilizando herramientas de *computer vision* para la detección de objetos y sus propiedades en la imagen. A partir de estos datos, se podrían generar preguntas de la forma:

- *What colour is the ball?*
- *How many objects are under the table?*
- *What has the girl in her hands?*

7.2.4. Migración de Infraestructura

Actualmente el *backend* de la aplicación se encuentra configurado en un servidor provisto de manera gratuita por la Facultad de Ingeniería. El mismo cumple con los requisitos mínimos para asegurar el correcto funcionamiento de la aplicación. Sin embargo, este servidor no puede ser accedido de manera remota para su mantenimiento y no está pensado para escalar a grandes cantidades de usuarios.

Dado que nuestra solución utiliza Docker [37] en su configuración, es posible instalar la aplicación en cualquier servidor que cuente con dicha herramienta y cumpla los requisitos de Hardware. Esta facilidad, permite que el servidor pueda ser ejecutado en la nube, en servicios tales como AWS [40] o Microsoft Azure [41], permitiendo así beneficios tales como:

- Mantenimiento remoto
- Hardware configurable a medida
- Alta disponibilidad
- Tolerancia a Fallos
- Seguridad
- Respaldos de la Base de Datos

Por estos motivos, consideramos que para que la aplicación sea utilizada en la práctica por maestros y alumnos de diferentes escuelas, es importante que a futuro la infraestructura se actualice.

7.2.5. Integración con proyectos anteriores

El proyecto se presenta como continuación de dos proyectos realizados a lo largo del 2018, donde se implementaron diferentes ejercicios y juegos para la enseñanza del inglés en las escuelas. Los mismos fueron realizados de manera independiente, por lo que actualmente para utilizarlos se requiere una instalación o acceso por separado.

Se propone la integración de dichos trabajos juntos con los realizados durante el 2019, con el fin de brindar una única plataforma a las escuelas, y así tener un

control centralizado de los datos y de la utilización de los proyectos. Además, esto permitiría utilizar una infraestructura común y una evolución en conjunto de todos los proyectos.

7.2.6. Generación de preguntas y respuestas en Inglés con técnicas de Aprendizaje Automático

Dado que este proyecto cuenta con la infraestructura y software necesario para almacenar los ejercicios generados por los maestros, consideramos interesante incentivar el uso del mismo para generar un corpus validado por los maestros de preguntas y respuestas correctas a partir de textos en inglés.

Una de las mayores restricciones del este proyecto al inicio fue la falta un corpus completo y lo suficientemente grande para utilizar técnicas de Aprendizaje Automático.

Dado que se puede esperar que se genere el corpus, el mismo puede utilizarse para hacer un trabajo similar a este pero utilizando las técnicas antes mencionadas.

8. Glosario

- **A1** Nivel de Inglés del Marco Común Europeo de Referencias Lingüísticas de la sección 9.2.
- **A2** Nivel de Inglés del Marco Común Europeo de Referencias Lingüísticas de la sección 9.2.
- **ANEP** La Administración Nacional de Educación Publica es el organismo estatal responsable de la planificación, gestión y administración del sistema educativo público en sus niveles de enseñanza inicial, primaria, secundaria, técnica y también en la formación docente terciaria en todo el territorio uruguayo.
- **API** *Application Programming Interface*. Es un conjunto de procedimientos y funciones que permiten la creación y el acceso a una aplicación u otro servicio.
- **AWS** *Amazon Web Services*, es una colección de servicios de computación en la nube pública (también llamados servicios web) que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por *Amazon.com*.
- **Backend** En el contexto de la separación entre una capa de presentación y una capa de acceso a datos. *Backend* refiere a la capa de la lógica y el acceso a datos, mientras que el término *Frontend* refiere a la capa de presentación.
- **Ceibalita** Computadora personal provista por el Plan Ceibal a los alumnos y maestros de todas las escuelas del Uruguay.
- **Código Abierto** El código abierto u *open source*, es el término con el cual se identifica al software distribuido y desarrollado bajo una licencia que permite a los usuarios el acceso al código fuente del software. El mismo puede ser modificado sin restricción en el uso y con la posibilidad de poder redistribuirlo (siempre y cuando sea bajo los términos y condiciones de la licencia con la cual fue adquirido el software original).

- **Dirección de Políticas Lingüísticas** Sección de ANEP cuya misión es formar ciudadanos plurilingües que puedan, por medio del uso de las lenguas, interactuar en ámbitos sociales, académicos y/o laborales.
- **Framework** Es una estructura conceptual y tecnológica de soporte definido, que sirve de base para la organización y desarrollo de software.
- **Frontend** Parte de la aplicación que expone la capa de presentación de la misma.
- **Movers** Nivel de Inglés dentro del nivel A1 del Marco Común Europeo de Referencias Lingüísticas de la sección 9.2
- **NER** *Named Entity Recognition* es la tarea que intenta localizar y clasificar las entidades nombradas mencionadas en un texto (sin estructurar), en categorías predefinidas. Por ejemplo: nombres de personas, organizaciones, lugares, valores monetarios, cantidades, etc.
- **NLTK** *Natural Language Toolkit* es un conjunto de bibliotecas y programas para el procesamiento del lenguaje natural para el lenguaje de programación Python.
- **Penn Treebank** Corpus que consiste en más de 4.5 millones de palabras de Inglés Americano. Sobre el mismo se realizó una anotación de *part-of-speech (POS)* siguiendo las *tags* propuestas en el Brown Corpus ([Francis 1964])[8].
- **PLN** El Procesamiento de Lenguaje Natural es la rama de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano.
- **POS Tagging** El *Part-of-Speech (POS) tagging* es la tarea de etiquetar palabras en un texto con su correspondiente etiqueta gramatical, según el contexto en el que se utilice la misma.
- **Python** Lenguaje de programación muy utilizado en la actualidad, para el cual existen una variedad de herramientas de PLN.

- **SPA** Single Page App, es una aplicación web de una sola pagina, es decir, la interacción de la aplicación es en una pagina. Tiene la característica principal de no tener que cargar los archivos múltiples veces, sino que solo cargarlos la primera vez.
- **SRL** *Semantic Role Labeling* es la tarea de etiquetar palabras o conjuntos de palabras en una oración, indicando su rol semántico. Por ejemplo: agente, objeto, etc.
- **Stanford NLP** Librería de Stanford para el Procesamiento de Lenguaje Natural disponible para el lenguaje Python, siendo esta una de las más importantes en el área. Abarca varias áreas del PLN, entre ellos el análisis de sentimientos, comprensión de frases, traducción automática, *parsing* de las oraciones, etc.
- **Starters** Nivel de Inglés anterior del nivel A1 del Marco Común Europeo de Referencias Lingüísticas de la sección 9.2
- **token** Es la mínima unidad del resultado del proceso de etiquetado (por ejemplo, palabras).
- **URL** URL son las siglas en inglés de Uniform Resource Locator. Es la dirección específica que se asigna a cada uno de los recursos disponibles en la red con la finalidad de que estos puedan ser localizados o identificados.

9. Anexo

9.1. Instalación y Ejecución de la Aplicación

Para instalar la aplicación (en cualquier sistema operativo), lo primer que hay que instalar es Docker. Para ello se pueden seguir las instrucciones del *link* oficial: https://docs.docker.com/get-docker/.

Luego de tener instalado y configurado Docker, se necesita instalar la herramienta *docker-compose*. Para ello, se pueden seguir las instrucciones oficiales del siguiente *link*:

https://docs.docker.com/compose/install/.

Una vez que se instalaron ambas herramientas, se podrá acceder a los siguientes *sin ninguna configuración adicional*:

- *Iniciar la aplicación con todos los servicios (Postgres, Backend, Frontend)*

```
$ sudo docker-compose up
```

- *Iniciar solo el backend de la aplicación*

```
$ sudo docker-compose run backend python manage.py runserver 0.0.0.0:8000
```

- *Iniciar solo el frontend de la aplicación*

```
$ sudo docker-compose run web yarn startf
```

- *Generar un build de la aplicación frontend para exponerlo como archivos estáticos (como cuando se hostea en AWS buacket)*

```
$ sudo docker-compose run web yarn build
```

- *Correr los tests de validación*

```
$ docker-compose run backend python manage.py totalresults --validate
```

- *Correr los tests de entrenamiento*

```
$ docker-compose run backend python manage.py totalresults
```

- *Correr los tests de POS tag*

```
$ docker-compose run backend python manage.py postagcmp
```

9.2. Niveles de Inglés

Para la realización de este trabajo se tuvieron en cuenta los niveles de inglés a los que apunta el trabajo final. En esta sección se especifican los diferentes niveles de inglés según el Marco Común Europeo de Referencias Lingüísticas (CEFR por sus siglas en inglés).

Las figuras 39 y 40 presentan la explicación de cada uno de los diferentes niveles de Inglés de acuerdo a lo establecido por la Universidad de Cambridge.

Nivel CEFR	Habilidades Auditivas	Habilidades del Habla	Habilidades de Lectura	Habilidades de Escritura
A1	Puede mantener conversaciones básicas acerca de hechos puntuales. Por ejemplo: “Where does your rabbit live?” “It lives in my garden.”	Puede ir a un supermercado en donde se muestran los productos y preguntar lo que quiera. Por ejemplo: “Can I have this drink, please?”'	Puede entender información simple de un compañero. Por ejemplo: “My name is Anita. I'm 16 and I go to school in Brazil”	Puede escribir un mensaje simple en el cual diga a dónde fue y en el tiempo que volverá. Por ejemplo: “Gone to school. Back at 5 p.m”
A2	Puede mantener conversaciones simples en las cuales exprese opiniones. Por ejemplo: “This looks like a good party.” “Yes, and everyone’s wearing funny clothes.”	Puede preguntar por lo que quiere e intercambiar información básica con otros clientes. Por ejemplo: “who was first in the queue?”	Puede entender cartas con descripciones simples de eventos, ideas y opiniones. Por ejemplo: “I am sad because it is raining”	Puede escribir una carta corta con información acerca de hechos básicos. Por ejemplo, su nombre, edad dónde vive, etc.

Figura 39: Niveles A1 y A2 de CEFR

Nivel CEFR	Habilidades Auditivas	Habilidades del Habla	Habilidades de Lectura	Habilidades de Escritura
B1	Puede mantener conversaciones por un tiempo razonable. Por ejemplo: “How was your camping holiday this year? Did you get washed away in all that rain?” “When we got there the campsite was closed because of flooding. But we were really lucky – the holiday company offered us a cottage instead for the same price.”	Puede ir a un supermercado y preguntar lo que quiera, por más que se muestre o no.	Puede entender cartas con opiniones personales.	Puede escribir cartas simples describiendo hechos y eventos.
B2	Puede mantener conversaciones de un espectro de temas. Por ejemplo, conversaciones acerca de eventos actuales de las noticias.	Puede negociar por lo que quiera y preguntar con efectividad por devolución o intercambio de un elemento.	Puede entender lo que se dice en una carta personal, incluso cuando se utiliza lenguaje coloquial.	Puede escribir cartas expresando opiniones y dando razones.
C1	Puede ser parte de conversaciones de un amplio espectro de temas con un a buena fluidez y una gran variedad de expresiones	Puede lidiar con transacciones sensibles y complejas.	Puede leer rápidamente para lidiar con un curso académico.	Puede escribir cartas de cualquier tema con buen nivel de expresividad y eficacia.

Figura 40: Niveles B1, B2 y C1 de CEFR

Referencias

[1] Inglés sin Límites, ANEP.
`https://www.anep.edu.uy/codicen/politicas-linguisticas/focus-on-first`
Último acceso: 28/03/2020.

[2] Ceibal en Inglés.
`https://ingles.ceibal.edu.uy/`
Último acceso: 28/03/2020.

[3] «Construcción de herramientas para soporte a la enseñaza de lenguas» Proyecto de Grado, 2018. Facultad de Ingeniería, Udelar.
`https://www.fing.edu.uy/inco/grupos/pln/prygrado/InformeANEPejercicios2018.pdf`
Último acceso: 21/02/2020.

[4] «Aplicaciones Lúdicas de Soporte a la Enseñanza de Lenguas» Proyecto de Grado, 2018. Facultad de Ingeniería, Udelar.
`http://www.fing.edu.uy/inco/grupos/pln/prygrado/InformeANEPjuegos2018.pdf`
Último acceso: 21/02/2020.

[5] Daniel Jurafsky & James H. Martin. 2008. *Speech and Language Processing (2nd Edition)*. Prentice Hall.

[6] Shah Alam. 2013. *Soft Computing Applications and Intelligent Systems*. Second International Multi-Conference on Artificial Intelligence Technology.

[7] Material del curso 2018 de Introducción al Procesamiento de Lenguaje Natural, Facultad de Ingeniería, Udelar.
`https://eva.fing.edu.uy/course/view.php?id=211`
Último acceso: 13/04/2020.

[8] Penn Treebank.
`https://catalog.ldc.upenn.edu/docs/LDC95T7/cl93.html`
Último acceso: 03/04/2020.

[9] Dependency Parsing, Stanford
https://nlp.stanford.edu/software/nndep.html
Último acceso: 15/04/2020.

[10] Dependency Grammar and Dependency Parsing, Joakim Nivre
https://cl.lingfil.uu.se/~nivre/docs/05133.pdf
Último acceso: 15/04/2020.

[11] Beatrice Primus. 2012. *Semantische Rollen*. Heidelberg: Universitätsverlag Winter.

[12] Sanford's Jurafsky slides for SRL
https://web.stanford.edu/~jurafsky/slp3/slides/22_SRL.pdf
Último acceso: 30/04/2020.

[13] Summary of Semantic Roles and Grammatical Relations
https://pages.uoregon.edu/tpayne/EG595/HO-Srs-and-GRs.pdf
Último acceso: 30/04/2020.

[14] Dependency Parsing, Stanford
https://www.aclweb.org/anthology/W03-0419.pdf
Último acceso: 28/04/2020.

[15] Janet Kourany. 2019. *Science and the Production of Ignorance: When the Quest for Knowledge Is Thwarted*. The MIT Press.

[16] Dinamic Programming, Stanford
https://web.stanford.edu/class/cs97si/04-dynamic-programming.pdf
Último acceso: 09/04/2020.

[17] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval, Cambridge University Press*. 2008. ISBN: 0521865719.
Último acceso: 15/04/2020.

[18] Li Deng, Yang Liu. 2018. *Deep Learning in Natural Languaje Processing*. Springer; 1st ed.

[19] AllenNLP: A Deep Semantic Natural Language Processing Platform
https://pdfs.semanticscholar.org/a550/2187140cdd98d76ae711973dbcdaf1fef46d.pdf?_ga=2.41592610.1885288457.1581279670-1715699031.1580770374
Último acceso: 27/03/2020.

[20] Wordnet
https://wordnet.princeton.edu/
Último acceso: 17/03/2020.

[21] Wordnet Interface NLTK
https://www.nltk.org/howto/wordnet.html
Último acceso: 19/03/2020.

[22] Daniel Jurafsky James H. Martin. WordNet: Word Relations, Senses, and Disambiguation
https://web.stanford.edu/~jurafsky/slp3/C.pdf
Último acceso: 18/03/2020.

[23] Diane Litman. 2016. *Natural Language Processing for Enhancing Teaching and Learning*. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence

[24] Rubel Das, Antariksha Ray, Souvik Mondal and Dipankar Das. 2016. *A Rule based Question Generation Framework to deal with Simple and Complex Sentences*. Conference on Advances in Computing, Communications and Informatics (ICACCI),

[25] Michael Heilman, Noah A. Smith. 2010. *Good Question! Statistical Ranking for Question Generation.* Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

[26] Miroslav Blasták and Viera Rozinajová, 2008. *Machine Learning Approach to the Process of Question Generation*. Slovak University of Technology in Bratislava.

[27] Wei CHEN, Gregory AIST and Jack MOSTOW. 2009. *Generating Questions Automatically from Informational Text*. Project LISTEN, School of Computer Science, Carnegie Mellon University.

[28] Jakob Nielsen. 1993. *Usability Engineering*. Published by Morgan Kaufmann, San Francisco.

[29] Formato de Exámenes de Cambridge
`https://www.cambridgeenglish.org/exams-and-tests/key/exam-format/`
Último acceso: 11/04/2020.

[30] Named Entity Analysis and Extraction with Uncommon Words
`https://arxiv.org/pdf/1810.06818.pdf`
Último acceso: 17/04/2020.

[31] Bag of words
`http://vision.stanford.edu/teaching/cs131_fall1718/files/14_BoW_bayes.pdf`
Último acceso: 10/04/2020.

[32] ESLfast short stories
`https://www.eslfast.com/kidsenglish/`
Último acceso: 17/04/2020.

[33] Bucket S3 de AWS
`https://aws.amazon.com/es/s3/`
Último acceso: 19/04/2020.

[34] Stanford NLP
`https://stanfordnlp.github.io/stanfordnlp/release_history.html`
Último acceso: 18/04/2020.

[35] Stanford Core NLP
`https://stanfordnlp.github.io/CoreNLP/index.html`
Último acceso: 27/03/2020.

[36] Flanagan, David. 2006. *JavaScript - The Definitive Guide*. 5th ed., O'Reilly, Sebastopol, CA.

[37] Herramienta Docker
`https://www.docker.com/`
Último acceso: 15/04/2020.

[38] Django Framework
https://www.djangoproject.com/
Último acceso: 15/04/2020.

[39] React JS framework
https://reactjs.org/
Último acceso: 15/04/2020.

[40] Amazon Web Services
https://aws.amazon.com/
Último acceso: 18/04/2020.

[41] Microsoft Azure
https://azure.microsoft.com/
Último acceso: 18/04/2020.